



Diseño, impresión, montaje y control de un manipulador robótico

Departamento de Sistemas y Automática, Universidad Carlos III de Madrid

Grado en Ingeniería Electrónica Industrial y Automática

Trabajo de Fin de Grado

Autor: Alejandro Aguilar Romero

Tutor: Alberto Valero Gómez

Fecha de presentación: Septiembre de 2012

Agradecimientos

El presente TFG se ha llevado a cabo con el cariño y apoyo de muchas personas sin cuya participación no habría sido posible realizar. Durante varios años, estas personas han contribuido a mi formación no solo profesional, sino también como persona y por ello, quiero expresarles mi gratitud por su paciencia y su confianza.

En primer lugar, me gustaría agradecer a mi familia todo el apoyo recibido, con especial mención a mi madre por animarme siempre que surgía algún problema, y a mi padre por proporcionar soluciones que por mi cuenta no se me hubieran ocurrido.

En segundo lugar, agradezco a mi compañero y amigo Mario Almagro toda la ayuda que, de manera altruista, me brindó durante la realización de este proyecto y también durante mi etapa universitaria.

Finalmente, doy las gracias a mi tutor Alberto Valero por introducirme en este mundo de las impresoras 3D y por estar disponible cuando le he necesitado.

Resumen

Este proyecto forma parte de un grupo de proyectos desarrollados en la Universidad Carlos III de Madrid cuyo objetivo es probar la viabilidad de la robótica imprimible de bajo coste. La principal herramienta utilizada es la impresora 3D Thing-O-Matic perteneciente al Departamento de Ingeniería de Sistemas y Automática, la cual ha supuesto un gran giro en el ámbito de la robótica.

Para probar la viabilidad de estos robots, se diseñará y construirá un brazo robótico modular, al que se le añadirán servomotores que le infieran el movimiento necesario. El software utilizado será OpenSource para que cualquier usuario tenga acceso a él y los elementos necesarios (servomotores, piezas...) serán de bajo coste.

Se pondrá especial atención a las posibles mejoras que se puedan incluir en los diferentes módulos y se asentarán las bases necesarias para que cualquier usuario pueda construirse un brazo robótico con fines educativos.

Por último, se implementará un software de control para comprobar de manera real la efectividad del robot imprimible completo, que será capaz de realizar movimientos en los tres ejes del espacio a través de un ordenador que servirá como punto de control.

Abstract

This project is part of a group of projects developed at the University Carlos III in Madrid, whose objective is to test the viability of the low cost printed robots. The main tool used is the 3D printer Thing-O-Matic, which belongs to the Department of Systems Engineering and Automation and that has led the field of robotics to a new level of innovation.

To test the viability of these robots, the author will design and build a robotic arm consisted of several modules with servomotors to add the required movement. The software needed will be OpenSource so that everyone has free access to it, and the needed elements to build the robot will be low cost.

The main reason of using different modules is to be able to modify and improve separately the robotic arm, laying the foundations for printing your own robot with educational purposes.

Finally, the author will implement a control software to check in real time the effectiveness of the designed robot, which must be able to move in the three-dimensional space through its control point, a computer.

ÍNDICE GENERAL

Agradecimientos	1
CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS	8
1.1 Objetivos	9
1.2 Marco del proyecto	9
1.3 Medios empleados	10
CAPÍTULO 2: ESTADO DEL ARTE	11
2.1 OpenSource	12
2.1.1 Filosofía de libre utilización y modificación	12
2.1.2 El proyecto Thingiverse	13
2.1.3 Openscad	14
2.1.4 ReplicatorG	15
2.1.5 Arduino	16
2.2 Impresoras 3D	18
2.2.1 Funcionamiento	18
2.2.2 Historia de la impresión en 3D	18
2.2.3 Aplicaciones	20
2.2.4 Proceso de impresión	21
CAPÍTULO 3: DISEÑO DEL BRAZO ROBÓTICO	22
3.1 Aspectos técnicos de un brazo robótico	23
3.1.1 Estructura	23
3.1.2 Grados de libertad	25
3.1.3 Par necesario: Servomotores Futaba S3003	25
3.1.4 Trucaje del servomotor	26
3.2 Elección de módulos	29
3.2.1 Unión entre los diferentes módulos	29
3.2.2 Base	30
3.2.3 Antebrazo	33
3.2.4 Brazo	36
3.2.5 Muñeca	39

3.2.6	Pinza	42
3.2.7	Resultado final.....	47
3.3	Cinemática directa	48
CAPÍTULO 4: PUESTA EN MARCHA.....		51
4.1	Electrónica.....	52
4.1.1	Diagrama de conexiones	54
4.1.2	Elementos necesarios.....	54
4.2	Alimentación	55
4.3	Control.....	56
4.3.1	Aspectos a controlar.....	56
4.3.2	Mapeado	59
4.4	Aspecto real del robot.....	61
4.5	Conclusiones.....	63
4.6	Mejoras y trabajos futuros.....	63
CAPÍTULO 5: PLANOS Y BIBLIOGRAFÍA		65
5.1	Planos de las piezas.....	66
5.2	Bibliografía	78

ÍNDICE DE FIGURAS

<i>Figura 1 Entorno de programación del software de Openscad.....</i>	<i>14</i>
<i>Figura 2 Ejemplos de diseños creados con el software de Openscad</i>	<i>15</i>
<i>Figura 3 Vista frontal y trasera de una placa Arduino Uno.....</i>	<i>17</i>
<i>Figura 4 Impresora 3D Thing-o-Matic (izquierda) y bobina de plástico (derecha).....</i>	<i>20</i>
<i>Figura 5 Scott Summit, cofundador de "Bespoke Innovations", con una prótesis de pierna (izquierda) y Charles Overy, fundador de LGM, con una maqueta de una vivienda (derecha)...</i>	<i>20</i>
<i>Figura 6 Tipos de articulaciones en un manipulador robótico</i>	<i>23</i>
<i>Figura 7 Esquema de rotaciones del brazo robótico</i>	<i>24</i>
<i>Figura 8 Señal PWM de control en un servomotor Futaba S3003</i>	<i>26</i>
<i>Figura 9 Soldadura de cable de control al servomotor</i>	<i>27</i>
<i>Figura 10 Agujero realizado en la tapa trasera.....</i>	<i>27</i>
<i>Figura 11 Resultado final de trucar el servomotor.....</i>	<i>28</i>
<i>Figura 12 Ejemplo real de la unión entre dos módulos mediante la corona del servo</i>	<i>29</i>
<i>Figura 13 Pieza que conecta con las pesas.....</i>	<i>30</i>
<i>Figura 14 Pieza del módulo de la base</i>	<i>31</i>
<i>Figura 15 Resultado final de la base</i>	<i>32</i>
<i>Figura 16 Pared del antebrazo</i>	<i>33</i>
<i>Figura 17 Puente del antebrazo</i>	<i>34</i>
<i>Figura 18 Resultado final del antebrazo</i>	<i>35</i>
<i>Figura 19 Pared del brazo</i>	<i>36</i>
<i>Figura 20 Puente del brazo.....</i>	<i>37</i>
<i>Figura 21 Resultado final del brazo.....</i>	<i>38</i>

<i>Figura 22 Pieza de la muñeca que conecta con la corona del servomotor</i>	<i>39</i>
<i>Figura 23 Cara trasera del puente de la muñeca</i>	<i>40</i>
<i>Figura 24 Cara frontal del puente de la muñeca.....</i>	<i>40</i>
<i>Figura 25 Resultado final de la muñeca</i>	<i>41</i>
<i>Figura 26 Base de la pinza.....</i>	<i>42</i>
<i>Figura 27 Engranaje de la pinza</i>	<i>43</i>
<i>Figura 28 Barra de giro de la pinza</i>	<i>44</i>
<i>Figura 29 Agarre de la pinza</i>	<i>45</i>
<i>Figura 30 Resultado final de la pinza</i>	<i>46</i>
<i>Figura 31 Resultado final del brazo completo.....</i>	<i>47</i>
<i>Figura 32 Cinemática directa e inversa</i>	<i>48</i>
<i>Figura 33 Cinemática del brazo robótico</i>	<i>49</i>
<i>Figura 34 Ejes de rotación y geometría.....</i>	<i>49</i>
<i>Figura 35 Sistemas de referencia finales.....</i>	<i>49</i>
<i>Figura 36 Diagrama de conexiones.....</i>	<i>54</i>
<i>Figura 37 Elementos de montaje del robot</i>	<i>54</i>
<i>Figura 38 Fuente de alimentación ATX.....</i>	<i>55</i>
<i>Figura 39 Flujograma 1</i>	<i>56</i>
<i>Figura 40 Flujograma 2</i>	<i>57</i>
<i>Figura 41 Flujograma 3</i>	<i>58</i>
<i>Figura 42 Flujograma 4</i>	<i>58</i>
<i>Figura 43 Flujograma 5</i>	<i>59</i>
<i>Figura 44 Aspecto real del brazo.....</i>	<i>61</i>
<i>Figura 45 Brazo con pinza</i>	<i>62</i>

CAPÍTULO 1:

INTRODUCCIÓN Y OBJETIVOS

1.1 Objetivos

El objetivo de este proyecto consiste en el diseño y puesta en marcha de un brazo robótico creado a partir de piezas impresas a través de una impresora 3D. En torno a este objetivo principal, se proponen los siguientes subobjetivos:

- Dotar a los diferentes módulos del brazo robótico de los actuadores necesarios para su movimiento.
- Controlar por ordenador las diferentes posiciones que el robot es capaz de alcanzar. Se implementará una solución para coger un objeto y posicionarlo en otro lugar. Además, se propone un programa de control que utilice teclas de acceso rápido del teclado como indicadores del movimiento (por ejemplo, la tecla ↑ indicará a un módulo a realizar un movimiento ascendente).
- Diseñar la estructura de manera que se le puedan añadir fácilmente mejoras como pueden ser otros módulos que permitan más movimientos debido al mayor número de grados de libertad.
- Realimentación de la posición y utilización de la misma para el movimiento del robot.

1.2 Marco del proyecto

La razón principal por la que elegí tomar parte en este proyecto reside en el auge del uso de impresoras 3D de bajo coste. En Mayo de 2011, el Departamento de Sistemas y Automática de la Universidad Carlos III de Madrid a la cual pertenece el autor, apostó por la compra del modelo Thing-o-Matic, uno de los últimos de la empresa Makerbot. Así, comenzaba una nueva concepción de la robótica: cada usuario podía imprimir sus piezas para robots con la única limitación de tu imaginación. Solo quedaba demostrar la viabilidad de estos robots imprimibles en aplicaciones más complejas.

El simple hecho de poder obtener de manera física aquellos objetos que antes solo podías imaginar, abrió un mundo de posibilidades. Se acabó el comprar kits de robots donde llanamente podías comprobar el funcionamiento del mismo mediante un software ya prediseñado. A partir de entonces, cada usuario podía involucrarse en el diseño y la construcción de sus propios robots, en lugar de gastarse una cantidad ingente de dinero en un modelo a pequeña escala.

Las reacciones no se hicieron esperar y pronto se creó un grupo de trabajo en la Universidad Carlos III de Madrid, donde bajo la supervisión de profesionales de la robótica como Alberto Valero (tutor de este proyecto), alumnos y profesores se sumergían en este apasionante universo aprendiendo de manera bilateral los detalles de esta tecnología. La iniciativa tuvo tanto éxito que en Octubre de 2011 (5 meses más tarde desde la compra de la primera impresora) se adquiría otra impresora del mismo modelo que la anterior para abastecer la demanda de impresión que los alumnos solicitaban. De esta manera, se comprobaba que esta tecnología era accesible para muchas personas y tenía el atractivo necesario para crecer de manera exponencial en los próximos meses.

Tras varios proyectos de carácter ilustrativo como la creación de pequeños robots móviles, se empezaron a concebir proyectos de mayor envergadura. Entre ellos, la replicación de la propia impresora 3D (Proyecto Clone Wars) para que cada alumno pudiera poseer una impresora de bajo coste en su casa o el proyecto sobre el cual trata esta memoria: la construcción de un brazo robótico.

A través de la realización de este proyecto, el autor ha experimentado todas las fases de desarrollo de un producto: desde los problemas en la fase de diseño hasta las dificultades de funcionamiento en la fase de experimentación. Asimismo, el autor ha ampliado muchos de sus conocimientos de robótica y electrónica.

Con todo, se puede añadir que el uso de impresoras 3D no solo mejora la creatividad y la innovación del usuario, sino que además supone una gran herramienta para la docencia.

1.3 Medios empleados

Para poder llevar a cabo este proyecto, ha sido necesario contar con los siguientes materiales y medios:

- Ordenador con sistema operativo Ubuntu.
- Acceso a la impresora 3D Thing-o-Matic de Makerbot junto con plástico de impresión.
- Placa Arduino Uno equipada con microcontrolador ATmega328.
- Cable de conexión USB-serie para comunicar la placa Arduino con el ordenador.
- Servomotores FUTABA S3003.
- Fuente de alimentación ATX para alimentación de los actuadores y de la placa Arduino.

Características: INPUT: AC 230V, 5A, 50 Hz

MAX OUTPUT POWER: 250W

DC OUTPUT: +5V---25A (Red), +12V---8A (Yellow), GROUND (Black)

CAPÍTULO 2:

ESTADO DEL ARTE

2.1 OpenSource

2.1.1 Filosofía de libre utilización y modificación

OpenSource es la filosofía basada en la libre utilización y modificación del código fuente, de manera que cualquier usuario puede acceder al contenido o adquirirlo de manera libre. Uno de los preceptos más importantes de esta filosofía es lo que en inglés se denomina *Free Redistribution*, es decir, libre redistribución.

Pero, ¿qué significa y qué consecuencias tiene compartir el código fuente? Cuando se compra o descarga un software determinado, se accede a la versión compilada, la que está lista para ser ejecutada en un ordenador y a través de la cual hacemos uso del software en cuestión. Esta versión, sin embargo, no se corresponde con la que el desarrollador creó (versión en código fuente), ya que ha pasado por un compilador que traduce este código para que el ordenador pueda interpretarlo. Esto provoca que sea extremadamente difícil modificar la versión compilada de la mayoría de las aplicaciones y casi imposible averiguar cómo el creador del software desarrolló las diferentes partes de su programa. La mayoría de fabricantes entienden esta consecuencia del proceso como una ventaja para evitar que otras compañías copien su código o lo utilizan como una ventaja competitiva para sus propios productos.

OpenSource se basa exactamente en lo contrario. El código fuente se incluye en la versión compilada y se motiva su modificación o personalización. Así, los desarrolladores del software entienden que, permitiendo la libre utilización del código, la aplicación será más útil y libre de errores a largo plazo.

Ahora bien, para poder ser considerado como OpenSource, han de cumplirse ciertos criterios:

- El programa ha de ser de libre distribución.
- El código fuente ha de estar incluido y podrá ser modificado por cualquiera.
- Las versiones modificadas pueden ser redistribuidas.
- La licencia no puede requerir la exclusión de otros software.

¿Y por qué está teniendo tanto auge esta iniciativa? La principal causa es la **rápida evolución** que experimentan los proyectos, pues habilitando la modificación del código por parte de cualquier usuario, cualquier comunidad de desarrolladores puede abarcar un trabajo de gran envergadura en muy poco tiempo. Además, se aportan **nuevas ideas** al modelo inicial debido al gran número de personas que participan en el proyecto.

Otra de las razones por las que la compartición de software está teniendo un éxito tan notorio es el aumento del **aprendizaje**. Incluso una persona con conocimientos básicos

en una materia puede participar en un proyecto de alto nivel junto a otras personas de más experiencia. La motivación en este caso no es el beneficio económico, sino la difusión de una idea por la que una comunidad siente pasión por desarrollar.

Ahora bien, debido a que la licencia anima a modificar y personalizar el código, es prácticamente imposible acceder a soporte técnico. Por ello, hay organizaciones que venden este software OpenSource como un producto con garantía y soporte técnico. Este ha sido un factor decisivo para aquellas empresas que se deciden por comprar este tipo de software (como por ejemplo la organización Red Hat Software con el sistema operativo Linux).

Las ideas del código abierto también se han extendido al ámbito del hardware, donde se comparten los archivos de diseño, los esquemas eléctricos, el firmware... Estas comunidades son muy jóvenes y tienen una gran proyección por delante, ya que se permite a los usuarios no solo modificar o estudiar los diseños, sino también construirlos y venderlos. En este caso, los límites residen en que todas las herramientas utilizadas en el diseño de un objeto o diseño sean accesibles por cualquiera (es decir, no usar software propietario).

Una vez dicho esto, se puede afirmar que todo el código que se ha creado en este proyecto es OpenSource puesto que tanto la electrónica como el software empleado están bajo licencia OpenSource.

2.1.2 El proyecto Thingiverse

Thingiverse es una página web dedicada a la recopilación y compartición de diseños digitales OpenSource. Entre las diferentes categorías existentes, encontramos la de impresiones 3D, donde se puede acceder a diseños de accesorios, extrusores, partes de impresoras 3D... etcétera.

De hecho, uno de los proyectos más interesantes de esta última sección es la impresión de una réplica de la máquina 3D (proyecto RepRap), lo que significaría conseguir tu propia impresora a través de piezas imprimidas por otra. También se pueden encontrar otros diseños menos mecánicos y más artísticos: desde maquetas de edificios emblemáticos hasta herramientas, pasando por pequeños electrodomésticos como lámparas.

En este caso, sin embargo, nos centraremos en la categoría de impresión 3D, pues todas las piezas diseñadas e impresas para este proyecto serán incluidas en esta, bajo el nombre de "Robotic Arm". Se puede acceder a los archivos a través de este enlace:

<http://www.thingiverse.com/thing:28969>

2.1.3 Openscad

Desde hace décadas, las herramientas de diseño asistido por ordenador (CAD, del inglés Computer-Aided Design) resultan indispensables para cualquier proyecto de ingeniería o arquitectura. El software más conocido en esta categoría es AutoCAD, pero desde hace pocos años han surgido alternativas con más limitaciones (menos funcionalidades), más intuitivas y dedicadas al diseño de piezas específicas manteniendo un coste bajo.

Entre estas alternativas se encuentra Openscad, el principal software que se ha usado para diseñar todas las piezas necesarias del brazo robótico de este proyecto. Este programa multiplataforma (compatible con Linux, Windows y MacOS) permite la creación de modelos 3D mediante operaciones sencillas como la suma o la resta de componentes básicos, típicamente cubos o cilindros. Mediante estas operaciones se pueden conseguir diseños de alta complejidad, centrándose en la renderización de los mismos para conseguir modelos parametrizables que posteriormente serán utilizados para su impresión mediante otro software del cual se hablará más adelante. Además, Openscad incluye dos técnicas para modelar: la primera se basa en la construcción sólida geométrica (CSG) y la segunda en la extrusión de bocetos 2D.

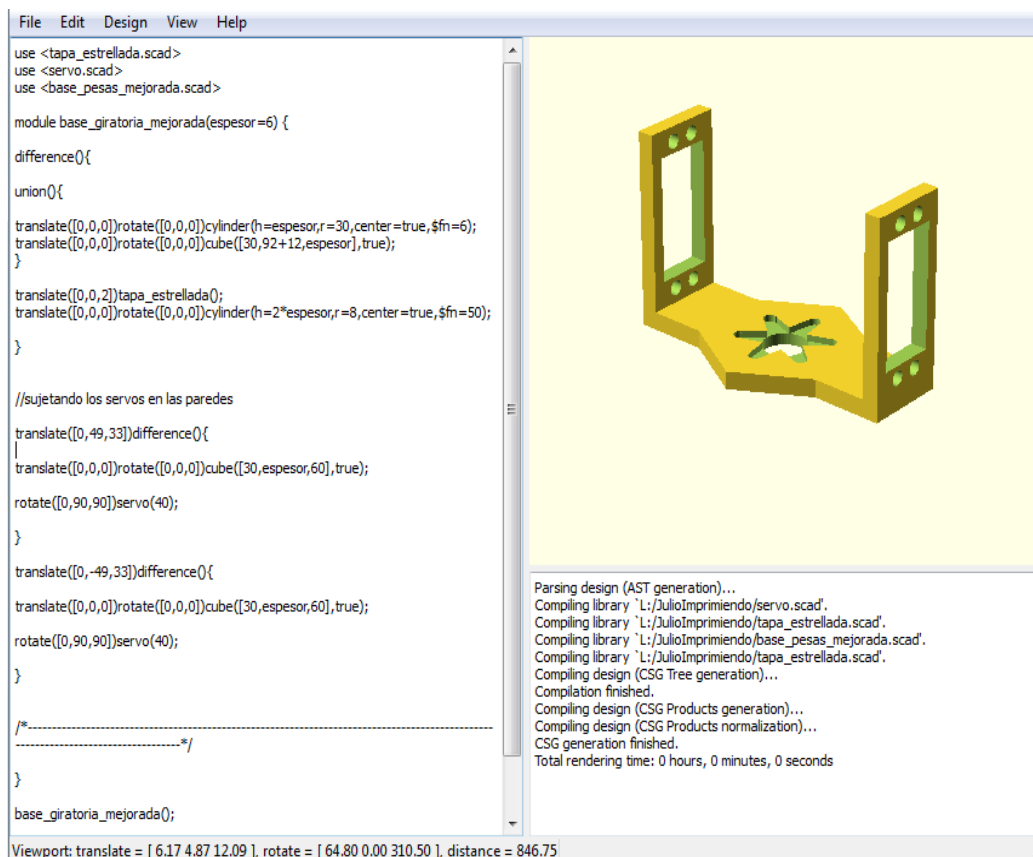


Figura 1 Entorno de programación del software de Openscad

A la hora de diseñar y a diferencia de otros programas como AutoCAD, Openscad permite la creación de diseños tridimensionales a partir de un script mediante el cual se indican las características de la pieza. El éxito y la clave de la rapidez y facilidad que predominan en este software es la posibilidad de dividir un problema de altas dimensiones en pequeños paquetes de tareas más simples.



Figura 2 Ejemplos de diseños creados con el software de Openscad

Una vez se ha programado el script y la pieza ha quedado definida, se guarda el diseño creado en formato ".scad". Este formato es exclusivo de Openscad, por lo que no es exportable a otras herramientas de diseño. Sin embargo, al renderizar el diseño, se puede exportar a otras plataformas mediante formatos como ".dxf" (AutoCAD, blender...), ".off", ".stl" o incluso como imagen.

En este caso, se utilizará el formato .stl para exportar la pieza e imprimirla en la máquina 3D. El ".stl" se corresponde a un archivo estándar de prototipado rápido y define las superficies que delimitan la forma del objeto. Este formato es de salida estándar para la mayor parte de los programas CAD, aunque los programas de modelado 3D no trabajan con estos archivos, por lo que hay que realizar una nueva conversión antes de poder obtener físicamente la pieza diseñada.

2.1.4 ReplicatorG

Este software es el que utilizan las impresoras 3D para transformar el formato ".stl" en un formato ".gcode" que permite la impresión. Estos archivos están definidos en un lenguaje de programación usado en control numérico (CNC), mediante el cual se le transmiten a la herramienta las diferentes instrucciones que permitirán la impresión del objeto. Entre las diferentes acciones se encuentran la trayectoria a seguir, la velocidad, la dirección... etcétera.

El formato ".gcode" será interpretado, por tanto, por la impresora 3D mediante el software de ReplicatorG y se obtendrá finalmente la pieza en formato físico.

2.1.5 Arduino

Arduino es una plataforma de electrónica abierta, lo que se corresponde con la categoría de Hardware OpenSource. De esta manera, las placas pueden ser fabricadas por uno mismo o compradas montadas de fábrica, pero el software siempre será descargado de manera gratuita. Además, debido a la licencia abierta del mismo, cada usuario podrá adaptarlo a sus necesidades.

El lenguaje de programación a utilizar es propio y existen infinidad de tutoriales sobre el montaje de la placa, ejemplos de control de actuadores y las diferentes librerías. Además, la comunidad de usuarios de Arduino ha creado un foro donde se proponen soluciones a los problemas de aquellos menos experimentados. De esta manera se ofrece un soporte técnico donde la propia comunidad de desarrollo se encarga de mejorar y afianzar los conceptos, promoviendo la difusión de esta plataforma.

Existen multitud de diferentes versiones de placas Arduino. Se diferencian entre sí en potencia, tamaño, pines de control e incluso aplicaciones (por ejemplo, la placa LilyPad está diseñada para aplicaciones sobre prendas). Para este proyecto se utilizará una placa Arduino Uno cuyas características son:

- Microcontrolador: ATmega328. Hoja de características en el anexo 1.
- Pines digitales de entrada/salida: 14 (6 de los cuales pueden ser usados como salidas PWM)
- Voltaje de operación: 5 V
- Voltaje de entrada recomendado: 7 - 12 V
- Voltaje de entrada: 6 - 20 V
- Pines analógicos: 6
- Intensidad DC por cada pin E/S: 40 mA
- Intensidad DC por cada pin 3.3V: 50 mA
- Memoria Flash: 32 kB (ATmega328)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Frecuencia de reloj: 16 mHz

Esta placa contiene todas las características para aprovechar al máximo la capacidad del microcontrolador. También dispone de una conexión USB, una conexión de alimentación y un botón de reset. El esquema eléctrico de la placa se adjunta en el anexo 2.

La alimentación de la placa se puede realizar tanto por conexión USB como de manera externa (adaptador AC-DC o pilas). La diferencia reside en que por USB, la intensidad que se obtiene es de 0.5 A, debido a la restricción del puerto USB, mientras que con una fuente de alimentación externa conectada al pin VIN, uno se asegura de que la intensidad recibida es la proporcionada por la fuente.

En cuanto a las entradas o salidas, es interesante comentar que cada uno de los 14 pines digitales disponibles trabaja a 5V. Los pines con funcionalidades más importantes son los siguientes:

- Serial (RX o TX): Usados para recibir (RX) o transmitir (TX).
- PWM (3, 5, 6, 9, 10, 11): Proveen salidas PWM de 8 bits necesarias para el control de los servomotores.

Por otro lado, cada uno de los 6 pines analógicos (A0 - A5) permite una resolución de 10 bits (1024 valores diferentes). Por defecto trabajan a 5V, aunque se puede cambiar su rango superior mediante el pin de AREF. Los pines con funcionalidades más importantes son los siguientes:

- AREF: Voltaje de referencia para entradas analógicas.
- Reset: Provoca el reset del microcontrolador.

Por último, Arduino permite su comunicación con otro ordenador, otra placa Arduino u otros microcontroladores. Esto se debe a que el ATmega328 habilita la comunicación serie UART TTL (5V), disponible también en los pines digitales o RX y TX. De esta manera, al conectar la placa mediante USB a un ordenador, los LEDs de RX y TX parpadearán cuando se reciba o transmita información. También se soporta la comunicación mediante I2C o SPI.



Figura 3 Vista frontal y trasera de una placa Arduino Uno

2.2 Impresoras 3D

2.2.1 Funcionamiento

La tecnología de impresión en 3D nace de la idea de crear objetos físicos a partir de un programa CAD en un ordenador. Esta tecnología, aparentemente fuera del alcance de muchos, ha ido evolucionando hasta tal punto que piezas de diseños complicados son ahora fácilmente imprimibles y replicables mediante una impresora 3D.

Pero, ¿cómo funciona una de estas impresoras? Bien, para empezar hay que entender el conjunto de tecnologías de fabricación que se incluyen con la impresión 3D. A este conjunto se le denomina "Additive manufacturing (AM)" y consiste en la creación de un objeto mediante la adición de material por capas. Esta idea no es tan artificial como parece, ya que, sin ir más lejos, podemos basarnos en el ejemplo de la formación de estalagmitas y estalactitas donde, a través de los años, se han ido depositando capas y capas de depósitos de mineral debido a las pequeñas corrientes de agua. La única diferencia entre este proceso y el de una impresora 3D es la rapidez y que el proceso se puede regular mediante un software de ordenador.

2.2.2 Historia de la impresión en 3D

La primera vez que se utilizó la fabricación por capas (AM) fue durante los últimos años de 1980 y los primeros años de 1990, como prototipado rápido (RP). Los prototipos permitían examinar el diseño físico del objeto y probarlo antes de empezar a producir de manera masiva el producto. El RP permitía obtener prototipos de una manera mucho más rápida que lo que se había utilizado hasta entonces, normalmente dentro de un margen de días o incluso horas después de concebir el diseño. Los diseñadores creaban el modelo en un software CAD y las máquinas seguían ese código para determinar cómo construir el objeto. Según se "imprimían" las diferentes capas cruzándose entre sí, nacía el concepto de impresión 3D.

Según transcurrían los años, se pulían los detalles de la impresión 3D, mejorando los tiempos de impresión, el tiempo que se necesitaba para terminar la pieza, la precisión... Los procesos ahora son más rápidos, los materiales y el equipo necesario es mucho más barato y cada vez más materiales están siendo adaptados para la impresión.

Al mismo tiempo que evolucionaba la técnica de la impresión por capas, también surgían importantes avances en las diferentes máquinas de impresión 3D. El origen de este tipo de impresoras es el proyecto RepRap iniciado por Adrian Bowyer en el año 2004 con el objetivo de desarrollar una máquina autorreplicante de código abierto. En mayo de 2007 el primer prototipo, llamado Darwin fue terminado y más tarde se logró la primera replicación. Desde entonces, la comunidad RepRap (máquinas originales

reap y diseños derivados) ha crecido de manera exponencial con una población estimada actual de alrededor de 4.500 máquinas. La segunda generación RepRap, llamada Mendel, fue terminada en Septiembre de 2009. Algunas de las principales ventajas de las impresoras de Mendel sobre Darwin son su mayor área de impresión, mayor eficiencia de los ejes, montaje más simple y sencillo, más barata y ligera y mayor portabilidad.

Inicialmente, Darwin y Mendel no fueron diseñados para el público en general, sino para las personas con algunos conocimientos técnicos. A medida que el proyecto RepRap fue adquiriendo carácter OpenSource, pequeñas empresas fueron apareciendo para comenzar a comercializar estas impresoras 3D, así como sus diseños derivados. La primera empresa fue MakerBot Industries, fabricando un primer lote de la Cupcake CNC en Abril de 2009. Sin embargo, aún quedaba un gran paso por dar y que incrementaría en gran manera la difusión de esta tecnología. El objetivo era encontrar un equilibrio entre la calidad y el precio, de manera que cualquier usuario pudiera tener su propia impresora en casa a un precio asequible y sin tener que recurrir a una inversión alta en una máquina profesional.

Con este motivo surgió la impresora Thing-O-Matic de la empresa MakerBot y a la cual ha tenido acceso el autor a través de la Universidad Carlos III de Madrid. Esta impresora pertenece a la familia de impresoras 3D de bajo coste y entre sus ventajas destacan:

- Bajo coste comparado a las profesionales (alrededor de 950€).
- Soporte técnico a través de la comunidad.
- Bajo coste de material (plástico).

Por otro lado, entre sus desventajas, podemos encontrar:

- Plataforma de impresión pequeña.
- Tiempos de impresión altos.
- Alta parametrización para conseguir calidades altas.

En la actualidad, el Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III de Madrid dispone de una MakerBot Thing-o-Matic para uso de los estudiantes. Dicha impresora fue totalmente montada por los alumnos y a día de hoy cualquier persona tiene libre acceso a ella. Su principal objetivo se define como el de estimular la imaginación y mejorar la creatividad de sus usuarios.



Figura 4 Impresora 3D Thing-o-Matic (izquierda) y bobina de plástico (derecha)

2.2.3 Aplicaciones

Los límites de esta tecnología aún están por determinar, puesto que desde las industrias de la automoción, médica e incluso aeroespacial hasta la del arte se han interesado por los resultados que ofrece. Esto se debe, sobre todo, al crecimiento exponencial que está experimentando debido a la gran cantidad de profesionales que han apostado por su utilización en diferentes campos.

Por ejemplo, en el campo de las prótesis está teniendo bastante auge debido al reducido coste y su alta fiabilidad. Igualmente, en el campo de la ingeniería se están produciendo enormes progresos en este aspecto, puesto que se realizan tests de fuerzas para experimentar con los diseños y luego extrapolar los resultados a tamaño real. De esta manera, el coste del prototipo es mucho menor, mientras que los resultados que ofrece son mucho más satisfactorios y verídicos. Por último, el campo de la impresión en 3D también tiene su aplicación en el arte y la moda. Diseñadores y artistas entienden este arte como una innovación para conseguir objetos con nuevos materiales y formas geométricas extravagantes y únicas.



Figura 5 Scott Summit, cofundador de "Bespoke Innovations", con una prótesis de pierna (izquierda) y Charles Overy, fundador de LGM, con una maqueta de una vivienda (derecha)

2.2.4 Proceso de impresión

Independientemente del modelo de impresora, el proceso de impresión generalmente sigue los siguientes pasos:

- Paso 1: CAD:
Creación del modelo 3D usando un software de ayuda por ordenador (CAD). El software provee de imágenes donde se puede comprobar la integridad estructural de lo que se espera sea el producto final, usando simulaciones bajo ciertas condiciones.
- Paso 2: Conversión a STL:
Se convierte el dibujo en CAD a formato STL. Este formato (acrónimo de standard tessellation language), es un formato de archivo desarrollado por "3D Systems" en 1987 para máquinas de estereolitografía.
- Paso 3: Transferir el STL a la máquina AM:
Se transmite el archivo STL al ordenador que está conectado a la impresora 3D. En él se determina el tamaño y la orientación de la pieza.
- Paso 4: Preparación de la máquina:
Se prepara la máquina con el plástico o cualquier otro material necesario para la impresión.
- Paso 5: Impresión:
Se espera a que la máquina imprima la pieza. El proceso es prácticamente automático. El grosor de cada capa es aproximadamente 0.1 mm, aunque se puede regular. Dependiendo de la máquina o los materiales usados, el proceso puede requerir más o menos horas.
- Paso 6: Obtención del objeto:
Se retira el objeto respetando un tiempo de espera para que la plataforma de impresión no siga calentando.
- Paso 7: Post-procesamiento:
Se perfecciona la pieza obtenida lijando los bordes o realizando cualquier otra acción necesaria para conseguir la pieza final. Esto puede incluir sumergir la pieza en agua para disolver plásticos solubles al agua.
- Paso 8: Aplicación:
Finalmente, se hace uso de la pieza.

CAPÍTULO 3:

DISEÑO DEL BRAZO ROBÓTICO

3.1 Aspectos técnicos de un brazo robótico

3.1.1 Estructura

Número de módulos. Alcance del robot.

Mecánicamente, un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. La constitución física de la mayor parte de los robots industriales guarda cierta similitud con la anatomía del brazo humano, por lo que, en ocasiones, para hacer referencia a los distintos elementos que componen el robot se emplean términos como cuerpo, brazo, codo y muñeca.

Los movimientos posibles para las articulaciones son: un desplazamiento (articulación de tipo prismático), un giro (articulación de rotación o de revolución), o una combinación de ambos, siendo éstos últimos menos habituales. Las dos primeras son las más usadas en la práctica. En nuestro caso, se utilizarán articulaciones de rotación, puesto que son las que más fácilmente se obtienen a partir de los actuadores seleccionados para este proyecto.

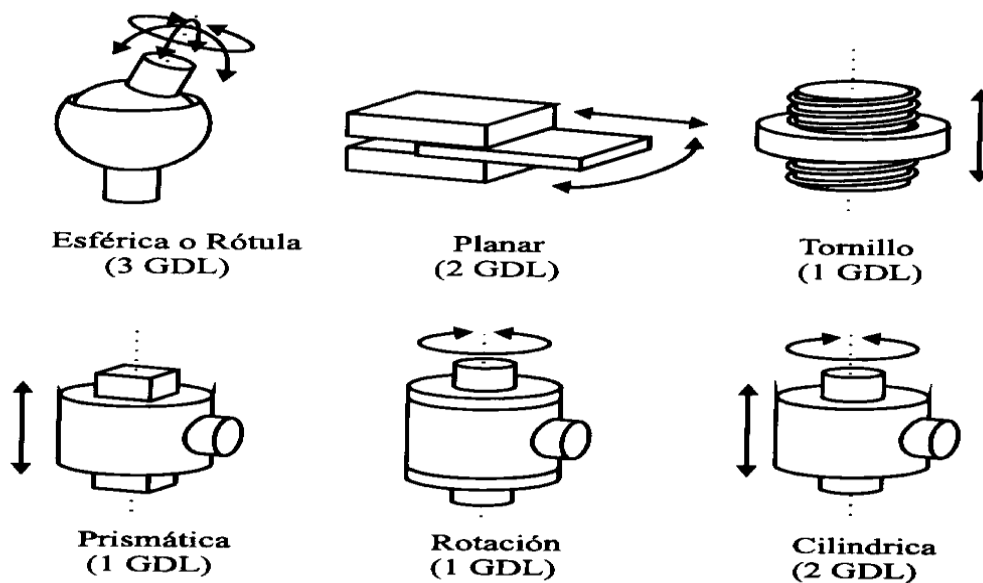


Figura 6 Tipos de articulaciones en un manipulador robótico

Al hablar de la estructura de un manipulador, es significativamente importante comentar el número de módulos que forman parte del mismo. El número de módulos afecta de manera directa al alcance del robot (zona de operación del manipulador), así como a las diferentes posiciones del espacio que puede alcanzar.

El sistema propuesto por el autor consiste en el siguiente número de módulos:

- Módulo base: Rotación en torno al eje Z.
- Módulo antebrazo: Rotación en torno al eje Y.
- Módulo brazo: Rotación en torno al eje Y.
- Módulo muñeca: Rotación en torno al eje Z.
- Elemento aprehensor: Pinza.

A continuación se muestra un esquema de los diferentes movimientos que permiten las articulaciones explicadas:

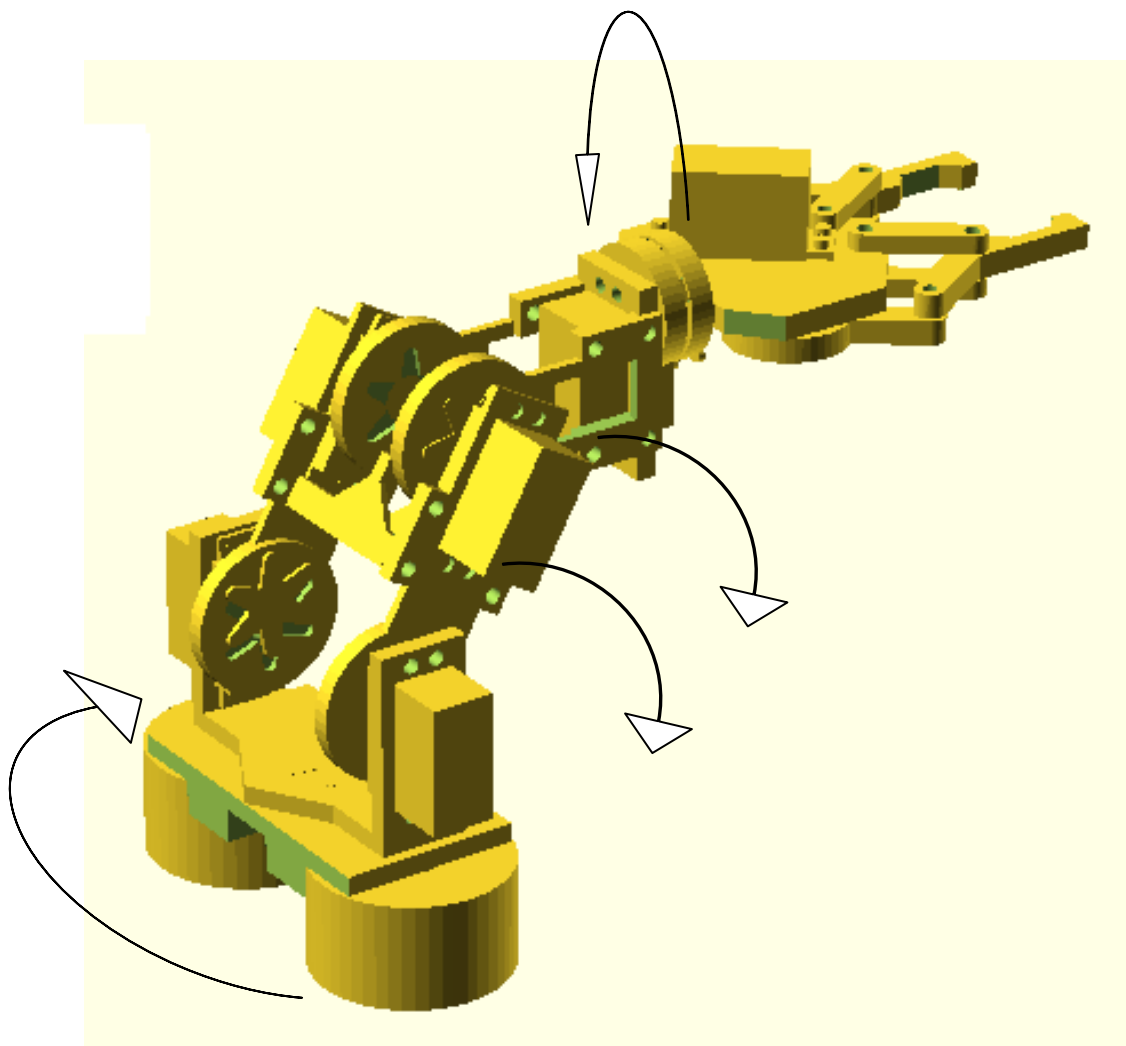


Figura 7 Esquema de rotaciones del brazo robótico

3.1.2 Grados de libertad

Cada uno de los movimientos independientes que puede realizar cada articulación con respecto a la anterior se denomina *grado de libertad* (GDL). El número de grados de libertad del robot viene dado por la suma de los grados de libertad de las articulaciones que lo componen. Puesto que las articulaciones más empleadas en la práctica son las de rotación y prismática, con un solo GDL cada una, el número de GDL del robot suele coincidir con el número de articulaciones de que se compone. En el caso del brazo del que se habla en este proyecto, **se dispone de 4 GDL** junto con el terminal de agarre.

La posibilidad de combinar articulaciones de un tipo u otro viene dada por la función que debe realizar el robot. Así por ejemplo, si se desea posicionar un objeto de cualquier manera en el espacio se requerirán 6 GDL (se debe tener en cuenta tanto la posición como la orientación). Sin embargo, con efectos ilustrativos, se ha decidido no dotar a nuestro brazo de los 6 GDL necesarios para alcanzar cualquier posición, en primer lugar por simplicidad y, en segundo, porque el objetivo principal de este proyecto es demostrar la funcionalidad del robot. Siendo esto así, siempre se podrían incluir más módulos una vez se haya comprobado el correcto funcionamiento del brazo.

3.1.3 Par necesario: Servomotores Futaba S3003

Los actuadores propuestos por el autor son servomotores Futaba S3003. Estos dotarán a los diferentes módulos del robot del par necesario para su movimiento. Los principales motivos de esta elección son el precio (son motores muy baratos), el tamaño (relativamente pequeños comparados con el par que ofrecen) y el rango de giro (180 grados, aunque se pueden modificar para conseguir un rango de giro mayor).

Las características técnicas de este tipo de motores son las siguientes:

- Velocidad: 0.23 seg/60 grados
- Par de salida: 3.2 Kg/cm
- Dimensiones: 40.4 x 19.8 x 36 mm
- Peso: 37.2 g
- Frecuencia PWM: 50 Hz
- Rango de giro: 180 grados

En cuanto al control de estos actuadores, este se basa en una señal PWM (Modulación por Ancho de Pulso) transmitida a través de un único pin del microcontrolador. Para posicionar el servomotor, hay que aplicar una señal periódica de 50Hz (20 ms de período), donde la anchura del pulso determina la posición del servo. Así, si la anchura es de 2.3ms, el servo se sitúa en un extremo y, si la anchura es de 0.3ms, se sitúa en el opuesto. Al dejar de enviar una señal, el servo entra en un estado de reposo en el que no se aplica fuerza para permanecer en esa posición, al contrario de lo que pasaba al enviar una señal de control.

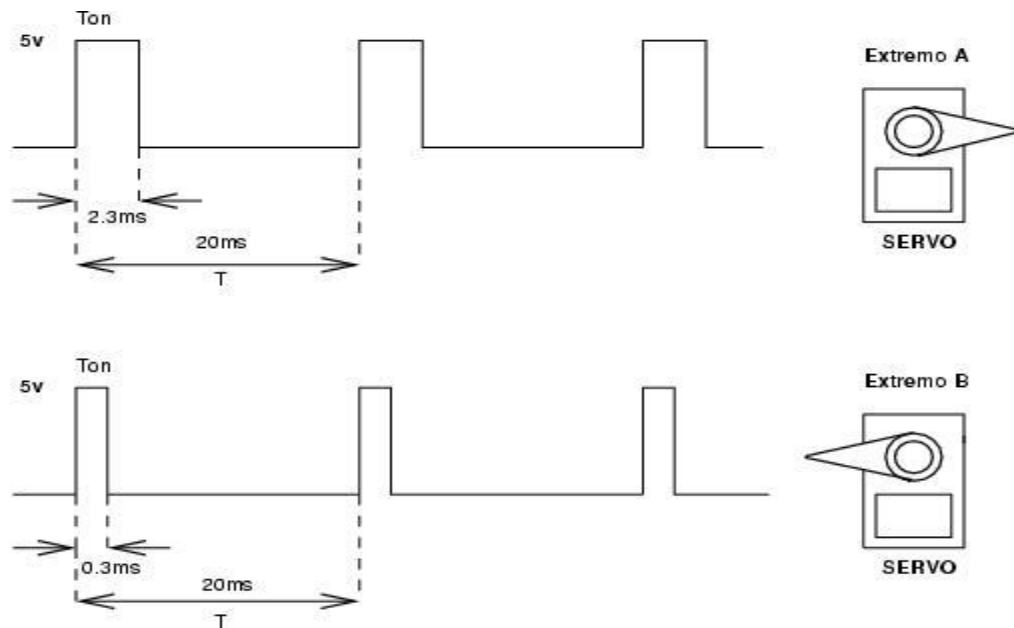


Figura 8 Señal PWM de control en un servomotor Futaba S3003

El servo dispone de un conector con tres cables:

- Cable rojo: Alimentación 4.5 - 6 V.
- Cable negro: Tierra.
- Cable blanco: Señal de control.

3.1.4 Trucaje del servomotor

En determinadas ocasiones, el servomotor no se utiliza únicamente para dotar de movimiento a un proyecto, sino que además se necesita un mayor ángulo de giro o un mayor control de la posición. Para aplicaciones como por ejemplo ruedas, es necesario lograr un ángulo de giro ilimitado (en lugar de los 180° originales proporcionados), por lo que se puede aplicar una incisión en el tope del eje del potenciómetro para conseguirlo. Esto habrá que tenerlo en cuenta a la hora de realizar el software. Sin embargo, para la aplicación del brazo robótico que nos ocupa, es de mucha ayuda contar con una señal de salida que indique la posición del módulo en el que está colocado el servo, ya que así se puede aplicar tensión únicamente cuando la pieza ya ha llegado a la posición que le indicamos en primer lugar.

Para conseguir esta señal, se soldará un cable a la parte trasera de la placa de control del motor, donde se encuentra una soldadura de la señal del potenciómetro (la cual nos dará la posición). Este cable se sacará por la tapa trasera e irá conectado directamente a una entrada analógica de la placa de arduino. Posteriormente se utilizará esta señal en el software de control del brazo.

Para soldar el cable se realizan los siguientes pasos:

- Se desatornilla la tapa trasera para acceder a la placa de control, donde veremos lo siguiente:

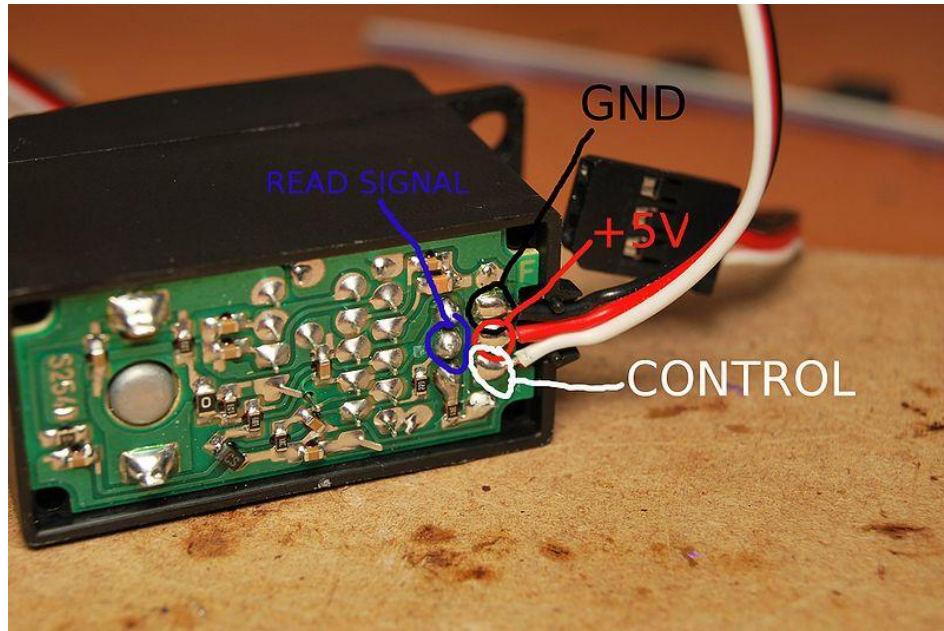


Figura 9 Soldadura de cable de control al servomotor

- Se suelda un cable en la posición de Read Signal, que es la señal del potenciómetro que nos indicará la posición en cada momento.
- Se realiza un agujero en la tapa trasera para poder extraer el cable sin afectar al conjunto del servo.



Figura 10 Agujero realizado en la tapa trasera

- Por último, se atornilla la tapa para obtener el resultado final:

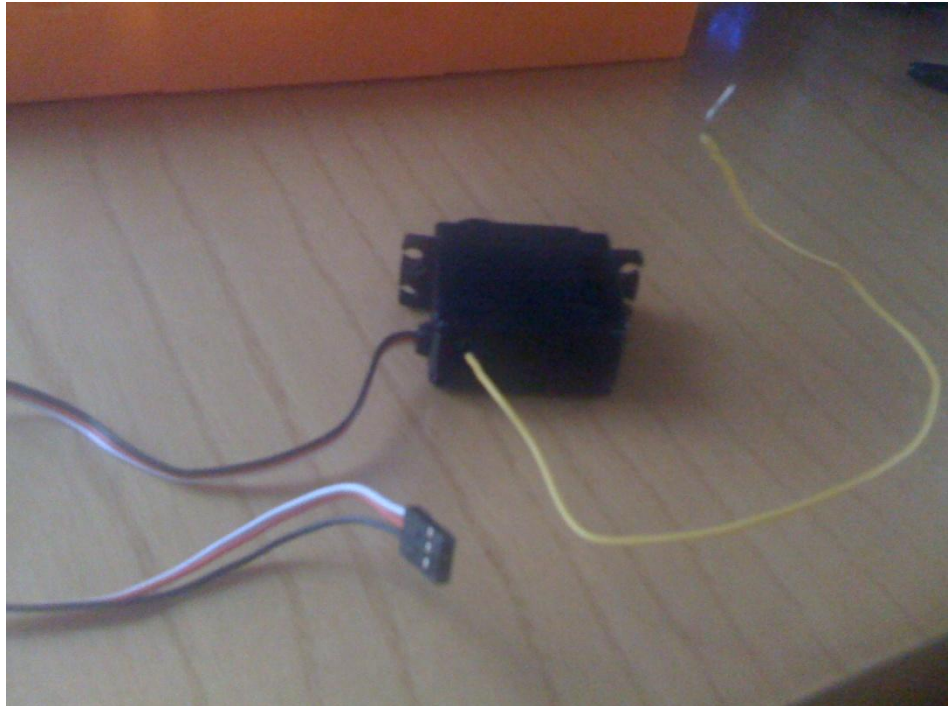


Figura 11 Resultado final de trucar el servomotor

Para realizar estos procedimientos hay que tener en cuenta las siguientes precauciones básicas:

- Al realizar la soldadura, no se debe aplicar demasiado calor a la placa, ya que podría dañarla de manera irreversible.
- La longitud de cable a soldar ha de ser suficiente para permitir todos los movimientos del brazo sin afectar al conexionado con la placa de Arduino.
- Puesto que los servos se alimentarán mediante fuente externa, no es necesario soldar cables de alimentación ni de tierra adicionales, aunque en otras aplicaciones pueden ser necesarios. En este caso, simplemente se soldarán a la misma altura que los cables de alimentación y tierra originales, poniendo especial atención a no soldar diferentes pines y al sobrecalentamiento de la placa (el rutado no puede alterarse).

3.2 Elección de módulos

3.2.1 Unión entre los diferentes módulos

El principal objetivo a conseguir es que mientras un servo se mantenga fijo en un módulo, el siguiente se mueva. Para ello, se utilizará siempre el mismo procedimiento: se realiza un agujero con el tamaño de la carcasa del servo en el módulo anterior y se realiza un agujero cilíndrico en la pieza siguiente junto con el hueco de una tapa estrellada. Esta forma se corresponde con una de las coronas del servo, de manera que se atornillará el motor a la pared del primer módulo y luego se apoyará la pieza (mediante el hueco cilíndrico) en la pared frontal del servo. Finalmente se atornilla la tapa estrellada.

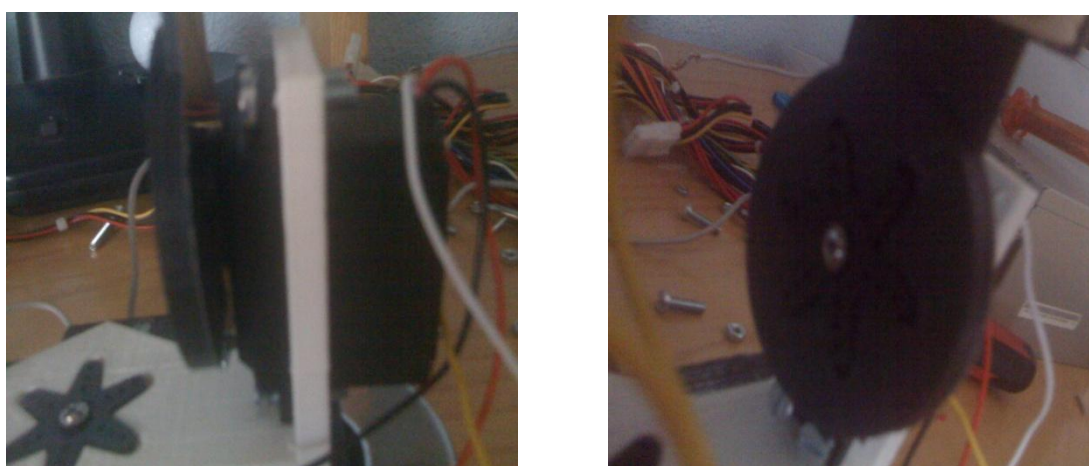


Figura 12 Ejemplo real de la unión entre dos módulos mediante la corona del servo

Este método tiene la ventaja de requerir muy poco material y de ser muy seguro, puesto que todo queda atornillado. Esto es válido también para la corona del servo, que debido a su forma estrellada, hace muy difícil que resbale en la pieza (al contrario de lo que pasa con las coronas circulares). Sin embargo, el inconveniente que tiene es que debido a la simetría del robot (dos piezas unidas por un puente en el brazo y antebrazo), se necesitarán dos servomotores en estos módulos para poder construir el brazo.

El hecho de incluir dos motores en el antebrazo y en el brazo no significa que ambos tengan que estar alimentados, pues esto provocaría sobrecalentamientos en los motores y pequeñas diferencias en el control (no tendrían por qué estar ambos en la misma posición siempre). **Por tanto, uno de los motores será el que transmita el movimiento y el otro estará vacío.** Este último dispondrá únicamente del mecanismo de ejes para permitir que la corona gire (y por lo tanto, se permita que el otro servo sea el que dirija el módulo siguiente).

3.2.2 Base

Base de sujeción con pesas

El módulo de la base ha de permitir el movimiento del brazo en torno al eje Z, por lo que necesita un servomotor colocado de tal manera que su eje de acción esté en ese eje. Además, hay que tener en cuenta que es en la base del robot donde hay que añadir más peso, puesto que esto reduce de manera considerable los pares estáticos y los momentos de inercia (especialmente en el extremo).

Por este motivo, se ha elegido añadir unas pesas de acero en la parte inferior de la base. Una vez se ha tenido en cuenta esta consideración, se ha diseñado una única pieza que irá sujeta a estas pesas y que contendrá un hueco para el servomotor en la posición en la cual el eje del mismo esté localizado en el eje Z. La elección del autor ha sido una única pieza sujeta a las pesas y que conecte el servomotor con la segunda pieza de la base que permitirá el giro del módulo.

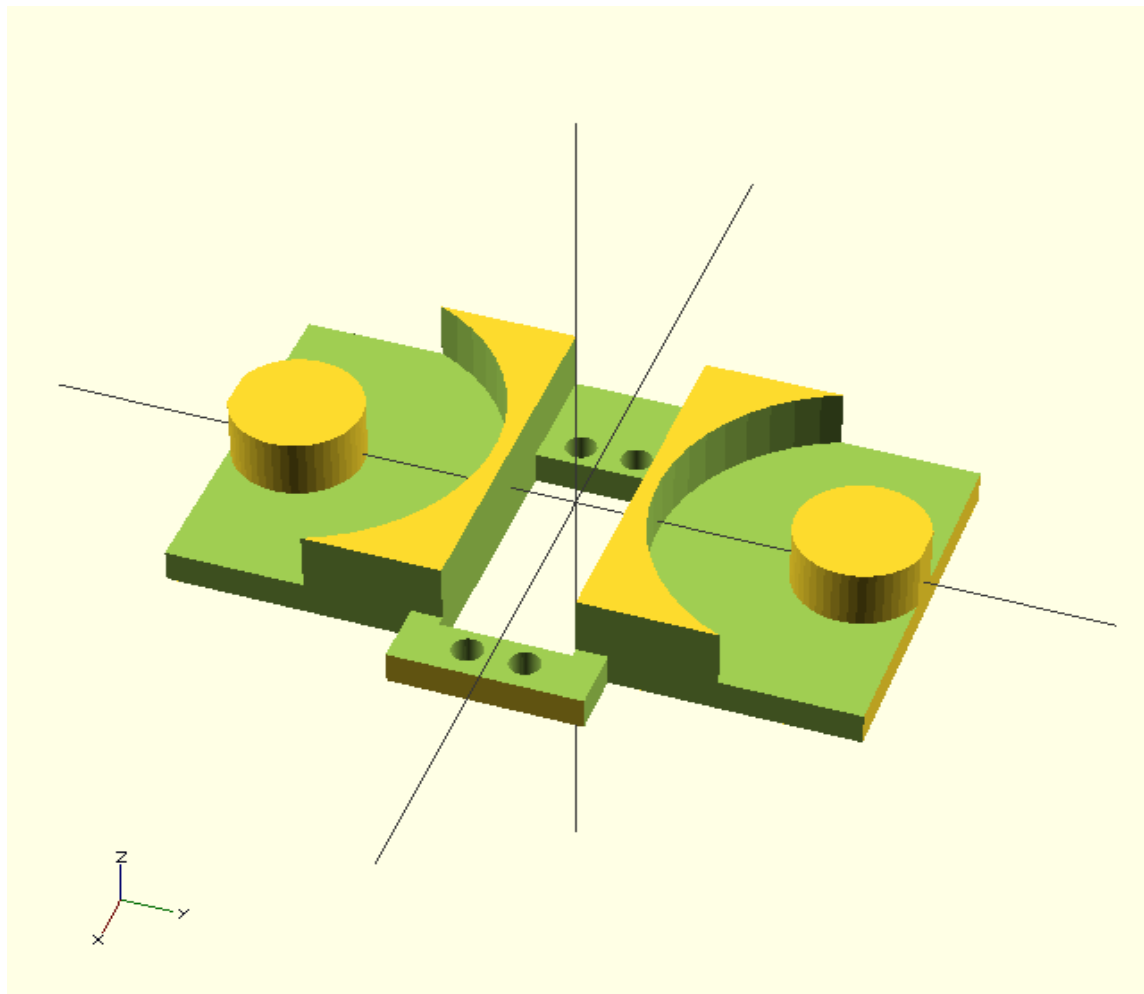


Figura 13 Pieza que conecta con las pesas

Base giratoria

En cuanto a la pieza que permite la rotación de la base, se ha escogido un espesor de 8 mm que impide que existan roturas en la pieza. Además, se ha diseñado en una única pieza sólida. El motivo por el cual el autor ha considerado construir una pieza en lugar de dividirla en partes más pequeñas es evitar las vibraciones y desajustes que pudieran existir a la hora de dotar movimiento a diferentes piezas unidas entre sí.

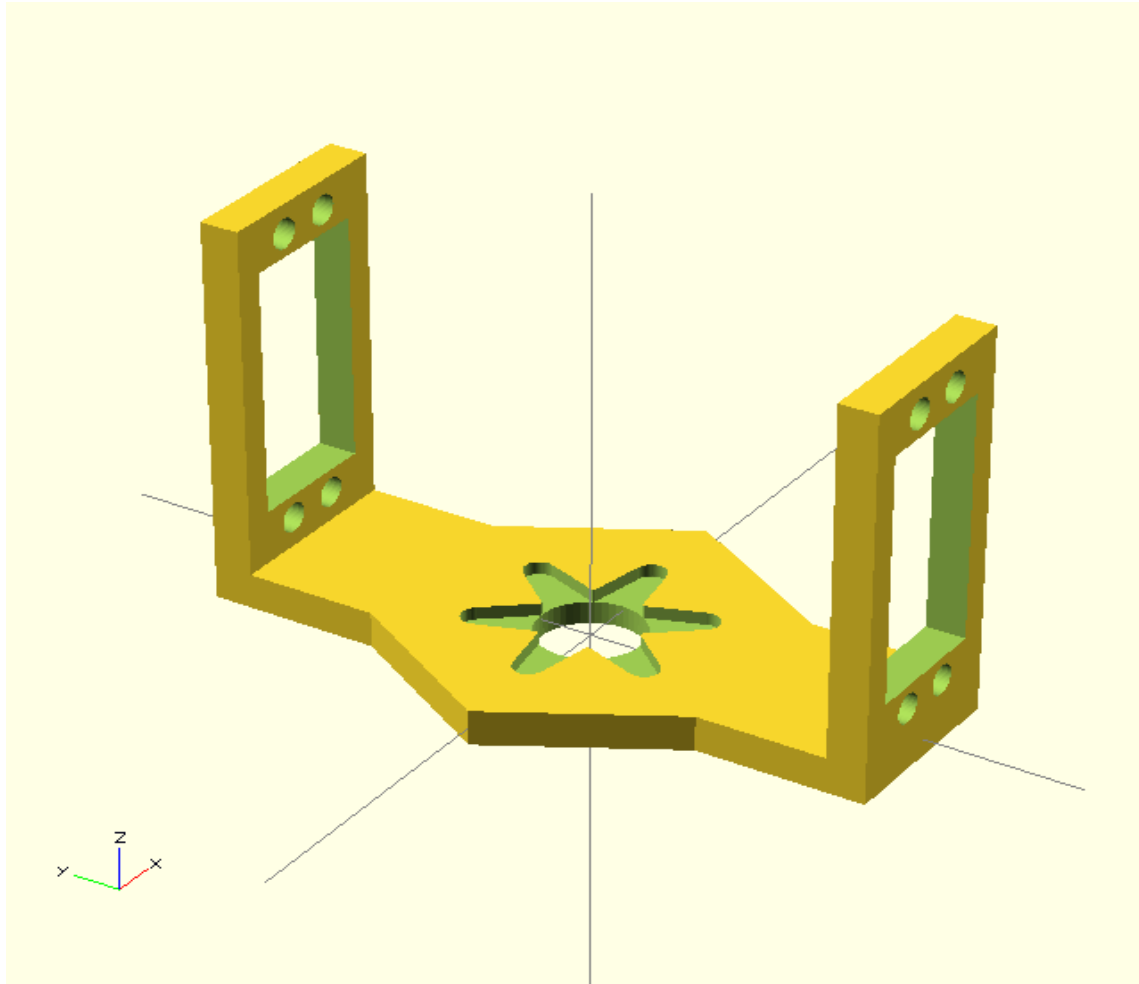


Figura 14 Pieza del módulo de la base

Hay que añadir que, teniendo en cuenta el método de unión de los módulos entre sí, hay que permitir que el eje de los servomotores que se incluyan en la base tiene que ser el que permita el movimiento en el siguiente. Es decir, puesto que el antebrazo ha de permitir la rotación en el eje Y, el eje de los servomotores ha de estar en ese mismo eje. Como se puede comprobar, los huecos de las paredes verticales donde irán colocados los servomotores tienen en esto en consideración.

Resultado final de la base

Una vez han quedado definidas las piezas de la base, el resultado final sería el que se muestra a continuación:

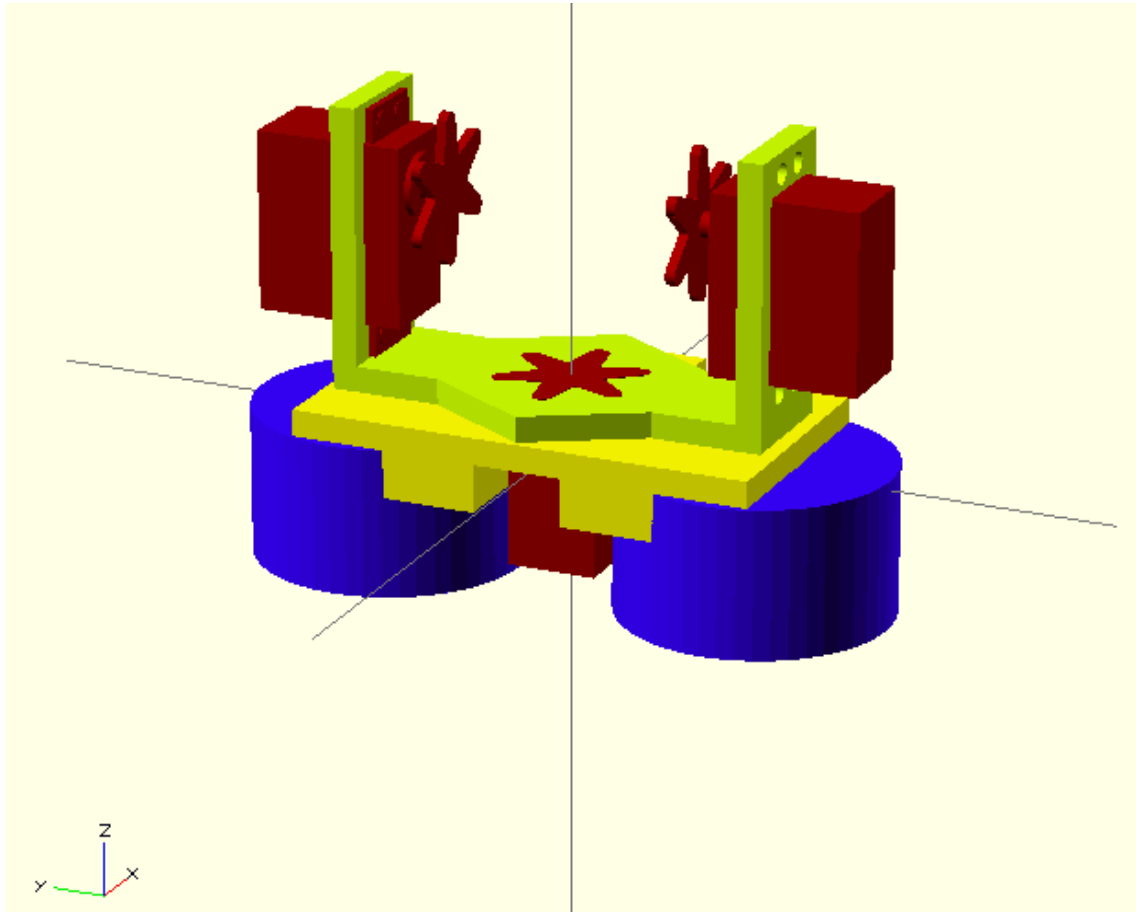


Figura 15 Resultado final de la base

Los objetos de color azul son las pesas cilíndricas que servirán de apoyo del robot. Estas encajarán con los huecos de la primera pieza de la base (pieza amarilla), la cual llevará un servomotor atornillado por la parte de abajo. Se han de usar tornillos sin cabeza (planos) puesto que si no, la pieza superior rozaría al rotar. Por último, la base giratoria (pieza verde) se sujetará sobre la pieza amarilla y rotará debido a la acción de la corona estrellada del servomotor.

La elección de las pesas en lugar de piezas impresas se basa en el hecho de que la mayoría de manipuladores tienen su base atornillada o sujeta a la plataforma donde está colocado. En este caso, el brazo no dispone de una plataforma de apoyo, por lo que se ha optado por esta solución. Sin embargo, si se dispusiera de otro entorno, bastaría con realizar el hueco de un servomotor en la nueva plataforma, sin necesidad de usar la pieza amarilla (base de sujeción con pesas).

3.2.3 Antebrazo

Pared del antebrazo

Para el diseño de esta pieza se han de tener en cuenta dos conceptos principales: en primer lugar, se ha de incluir el hueco de la tapa estrellada para poder conectar con el módulo inferior; en segundo lugar, se ha de incluir un hueco para un servomotor para poder continuar con la estructura. Una vez se han afianzado estos conceptos, se le añade altura a la pared para contar con un alcance mayor del robot.

Sin embargo, al aumentar la altura de la pared, podría haber desequilibrio cuando se realizara el movimiento. Es por esa razón por la que se ha de incluir una pieza adicional que conecte ambas paredes a la vez que sirva de sujeción.

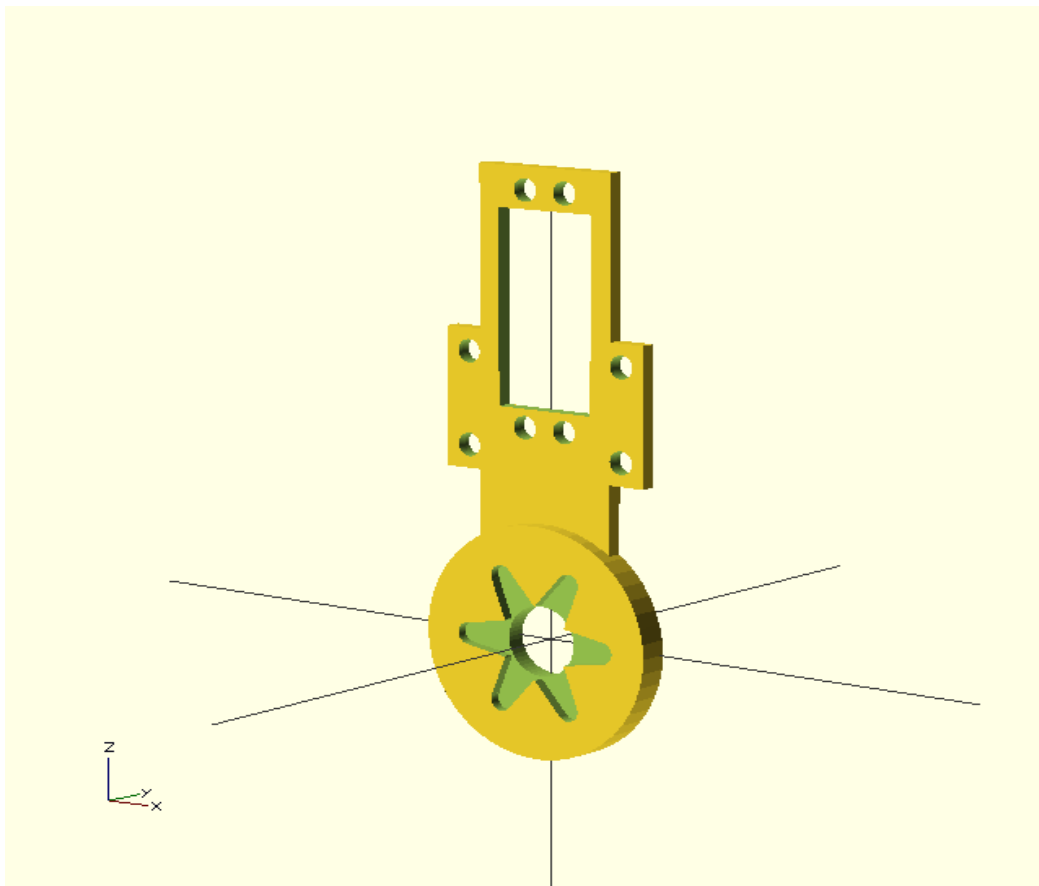


Figura 16 Pared del antebrazo

Como se puede observar en la figura 3.7, se ha añadido una pequeña plataforma con 4 huecos para atornillar el puente. Además, se ha minimizado el espesor en la zona superior, quedando solo la parte gruesa de la pieza para acomodar la corona del servomotor, reduciendo así el peso de este módulo.

Puente del antebrazo

Teniendo presente la colocación de los tornillos, se diseña una pieza con la longitud apropiada, midiendo la distancia entre las dos paredes del módulo de la base giratoria (hay que tener en cuenta la colocación de las coronas). Los servomotores irán colocados en los extremos de esta pieza, por lo que su parte alta deberá estar vacía. Para darle más consistencia a la pieza, se ha elegido un pequeño soporte cuadrangular que se une con las barras de sujeción.

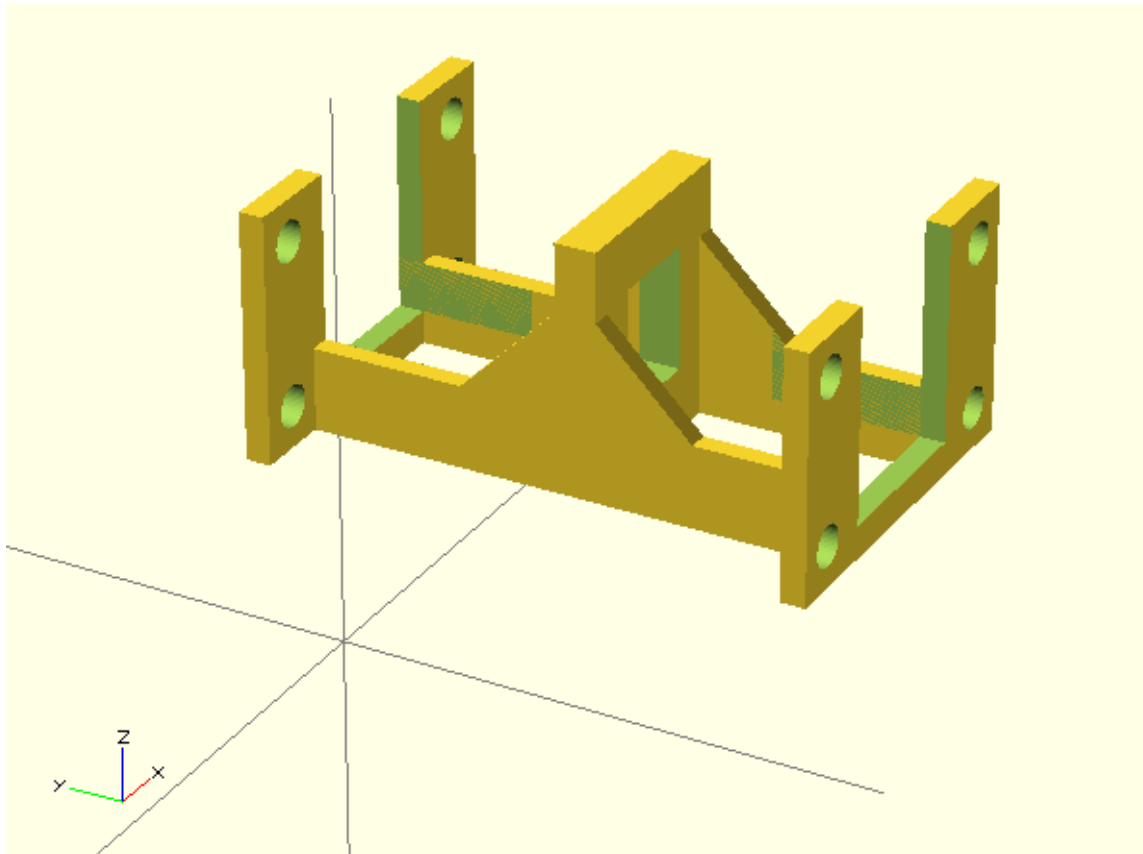


Figura 17 Puente del antebrazo

Por último, se rebaja material de donde sea posible para conseguir un tiempo de impresión de la pieza menor y abaratar el coste, además de conseguir su optimización. En cualquier caso, es una pieza de refuerzo y unión entre las dos paredes del antebrazo, por lo que lo principal es que la longitud sea la necesaria para que las paredes estén rectas con respecto a la posición de las coronas de los servos del módulo anterior.

Resultado final del antebrazo

El total del antebrazo estará formado por dos paredes situadas simétricamente entre sí y unidas con el puente.

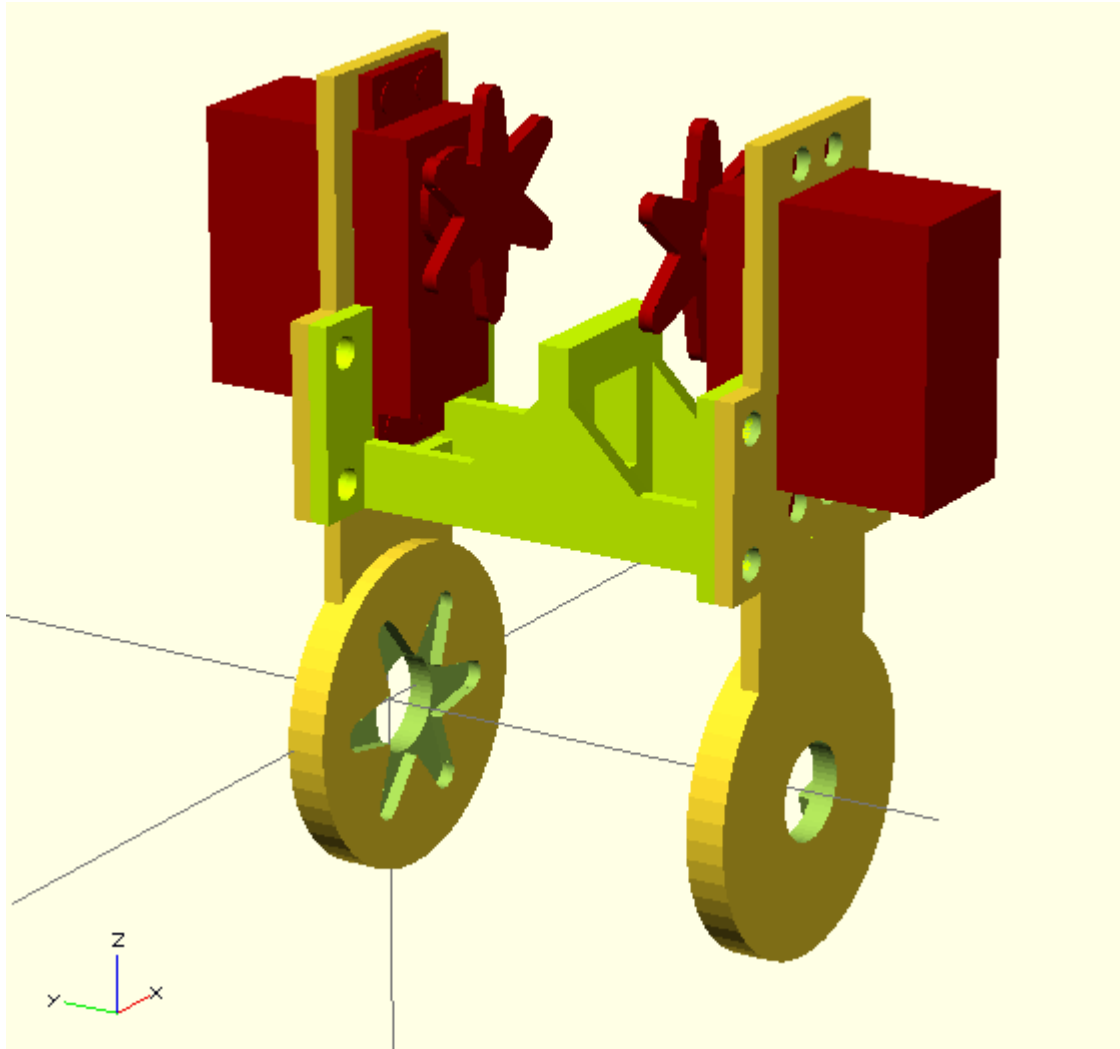


Figura 18 Resultado final del antebrazo

Nótese que, debido a la proximidad de las coronas de los servomotores (y por consiguiente de las siguientes piezas, puesto que al menos una parte de las mismas irán rodeando a la corona), el puente no tiene una barra de sujeción por la parte de arriba, lo cual le daría más consistencia al puente, pero sería una solución inviable.

3.2.4 Brazo

Hasta ahora se han diseñado dos de los módulos más importantes: la base y el antebrazo. Sin embargo, con estos módulos, el robot solo tiene dos grados de libertad (la base permite rotación en Z y el antebrazo rotación en Y), por lo que es necesario al menos la adición de un módulo que dote al sistema de más movimientos posibles. Con esta intención, se diseña un módulo de brazo que será muy similar al del antebrazo, excepto por lo hecho de que, al no continuar con más módulos, no será necesario incluir el hueco de los servomotores (aún así, se le incluirá un hueco para rebajar material).

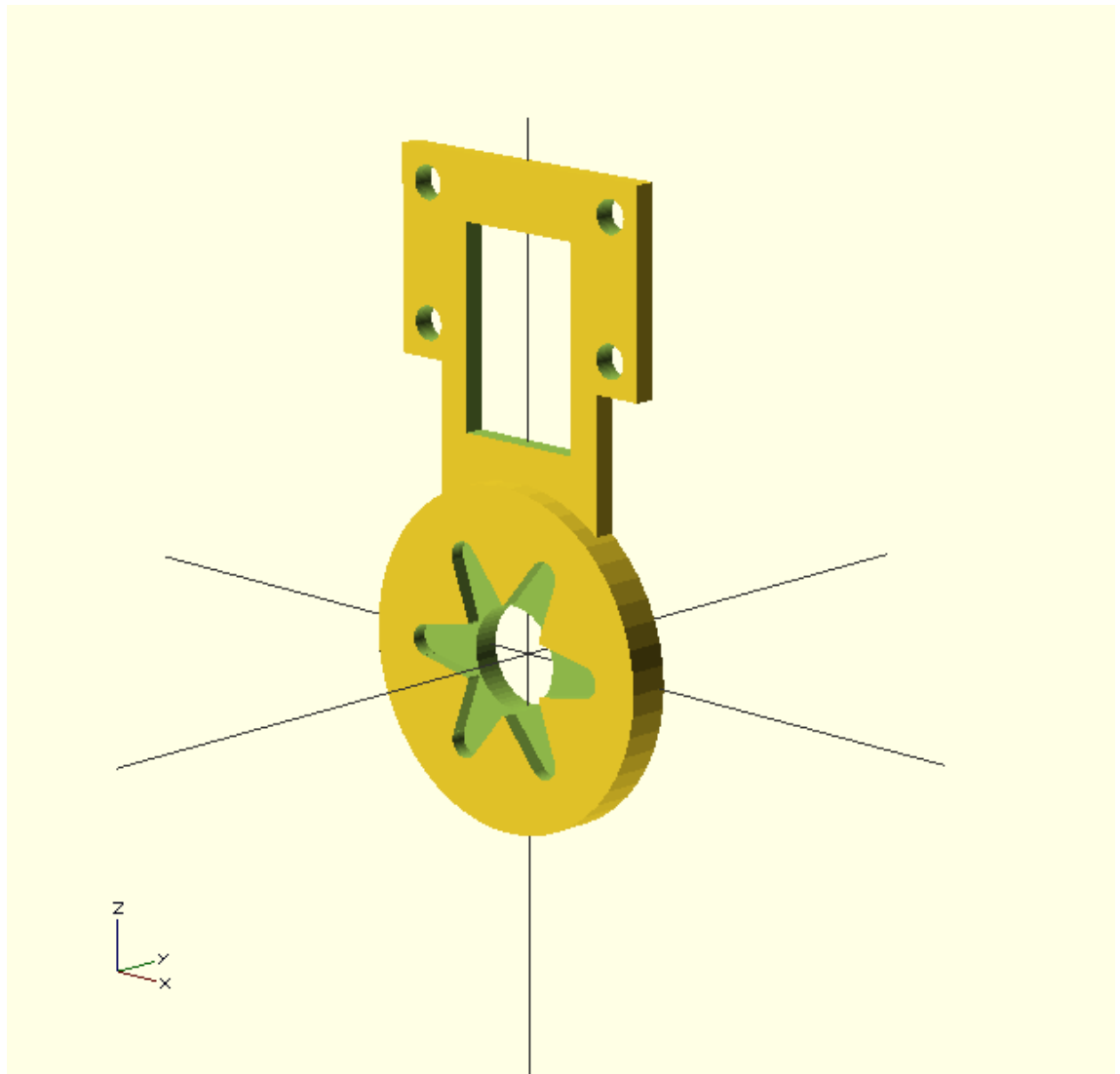


Figura 19 Pared del brazo

Sí será necesario incluir un puente que una las dos paredes y que, además, permite al siguiente módulo un movimiento en torno al eje Z. Por tanto, se necesitará un servomotor en esa posición.

Puente del brazo

De la misma manera que el puente del antebrazo, irá atornillado mediante los huecos diseñados en la pared del brazo. Como ya se ha explicado, tendrá un hueco para un servomotor cuyo eje estará situado en el eje Z y permitirá la rotación del elemento aprehensor, en este caso una pinza convencional.

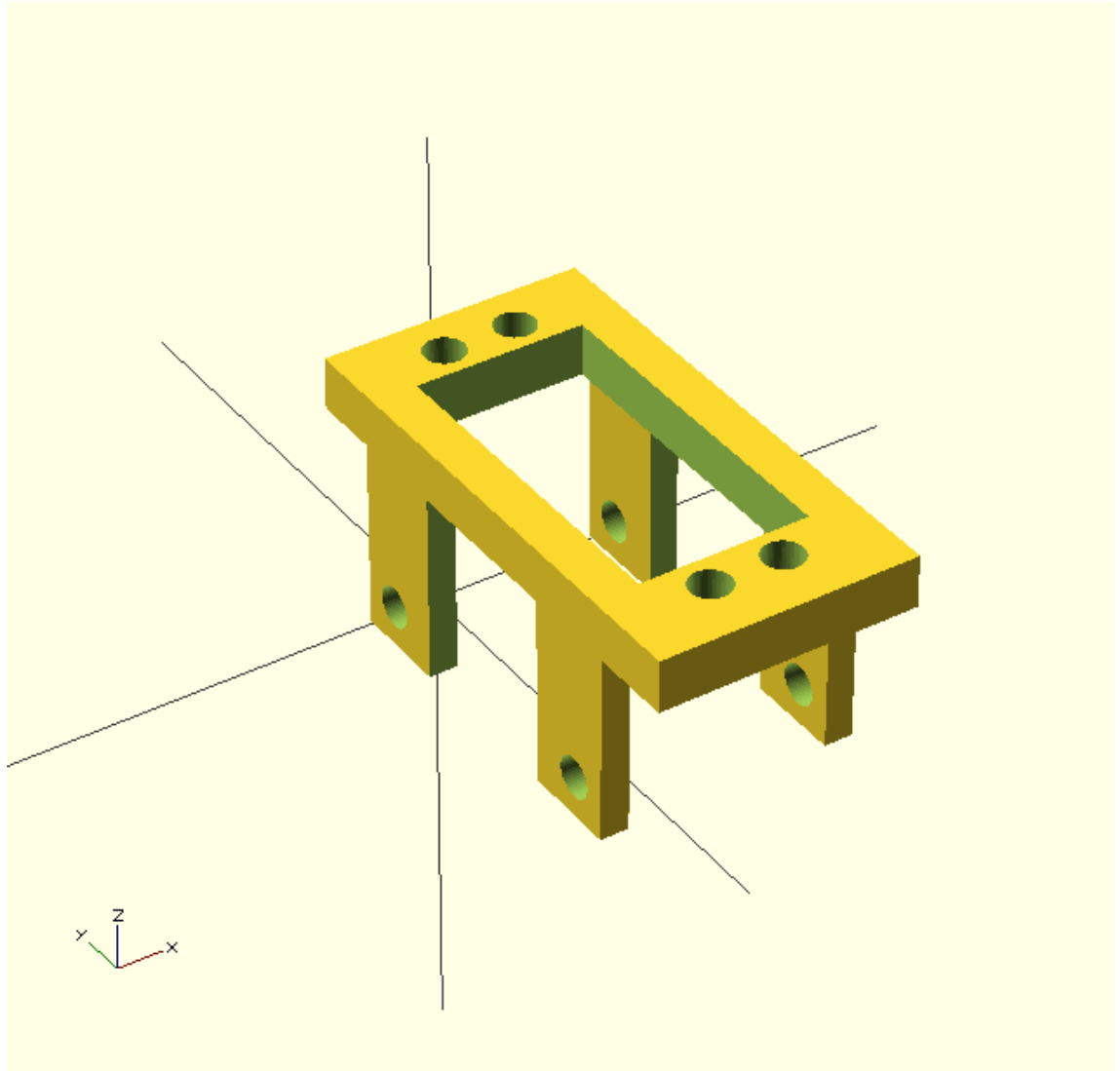


Figura 20 Puente del brazo

La principal consideración a tener en cuenta en esta pieza es que al incluir el servo, la sujeción de esta pieza vendrá dada por tornillos situados por debajo de la caja del servo, de manera que se han colocado los agujeros por debajo de la posición de la tapadera del motor.

Resultado final del brazo

El resultado final de este módulo quedaría como sigue:

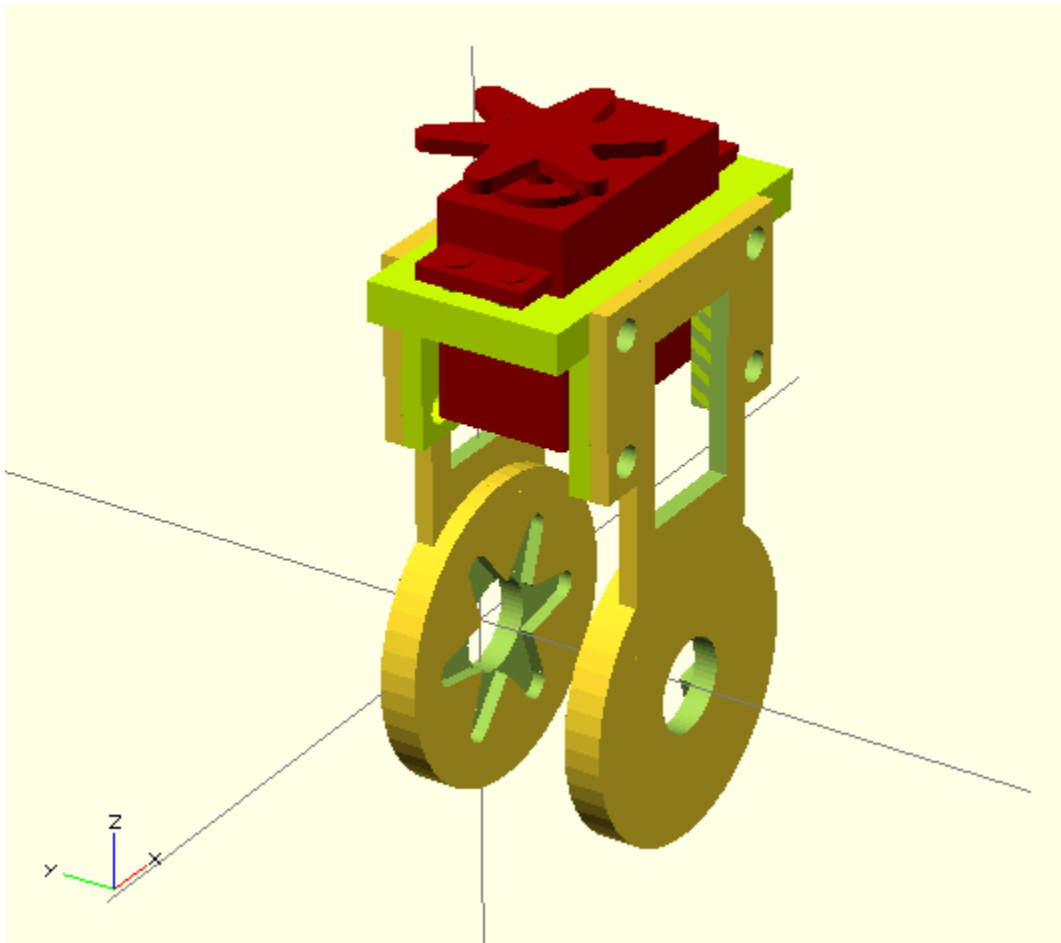


Figura 21 Resultado final del brazo

Se puede comprobar cómo, debido a la escasa separación entre ambas paredes, este módulo no necesita de un puente adicional, puesto que la pieza donde va colocada el servomotor es más que suficiente sujeción.

Es interesante ver que a partir de ahora, todo el peso de los módulos y piezas siguientes serán soportados por este servomotor y principalmente por su corona (ya que es el elemento de unión de módulos), razón por la que el autor decide no añadir más módulos. A partir de ahora, las piezas siguientes irán diseñadas en base a la inclusión de la pinza.

3.2.5 Muñeca

Este módulo solo ha de permitir un giro adicional en Z, por lo que podría parecer que simplemente el acoplamiento de la pinza sería suficiente para dotar de este movimiento. Lo único que haría falta sería colocar una pieza con el hueco de la corona del servomotor en el eje Z, mientras que el resto de la pinza tendría su eje en Y. Esta pieza, desafortunadamente, no es realizable con una impresora 3D a no ser que se le realicen bastantes ajustes y, aún así, se tardaría mucho tiempo en imprimir.

Por esta razón, el autor decide usar primero una pieza que se ajuste con la corona del servomotor. Posteriormente, otra pieza será el "puente" entre esta pieza y la base de la pinza.

Unión servomotor - puente

Una vez se tiene claro lo que se va a hacer, se pasa a la etapa de diseño de las piezas. En este caso, utilizaremos unos pequeños salientes en cada pieza para encajarla con la siguiente, tal y como se puede apreciar en la siguiente imagen:

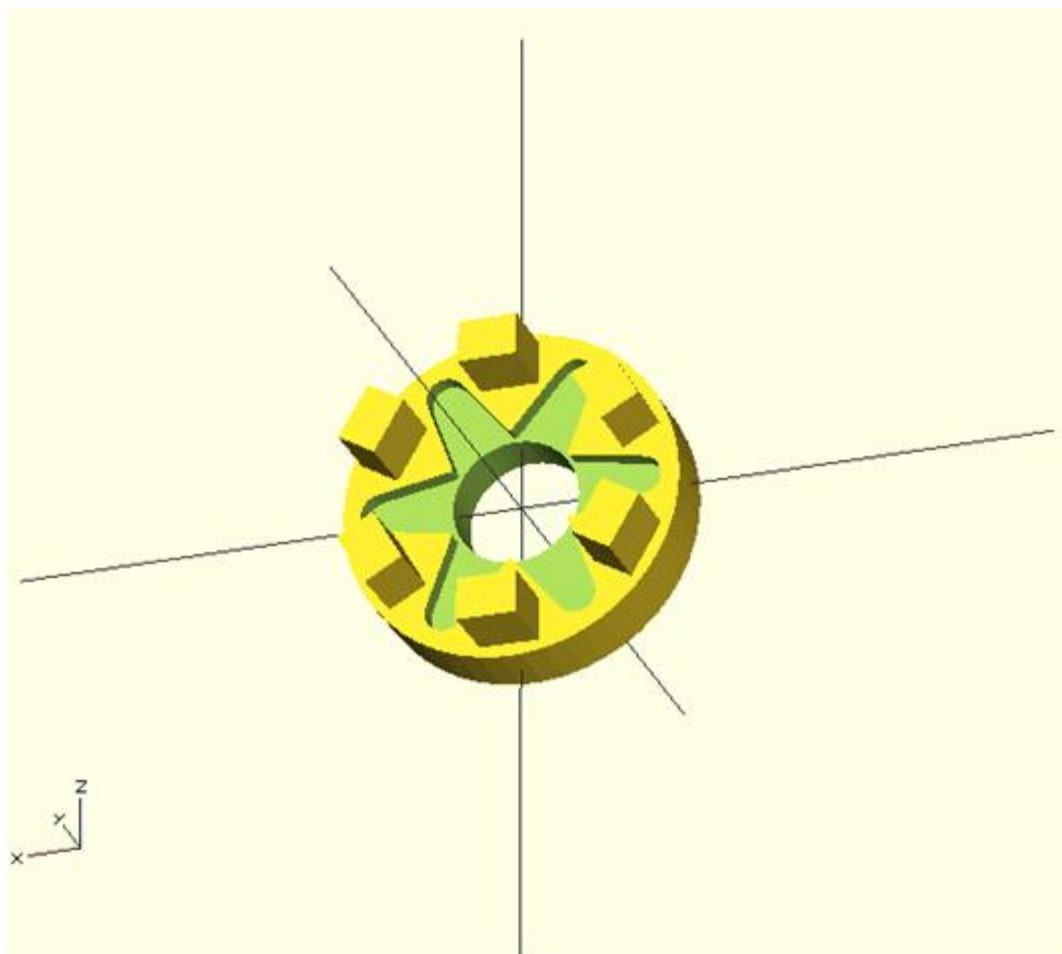


Figura 22 Pieza de la muñeca que conecta con la corona del servomotor

Puente de la muñeca

Ya se ha comentado que esta pieza permitirá la conexión entre la pinza y la pieza que engancha con la corona del servo. Para conseguirlo, solo se han de añadir los huecos correspondientes a los enganches de la pieza anterior por un lado mientras que, por el otro, se diseñarán otros salientes que a su vez conectarán con la base de la pinza.

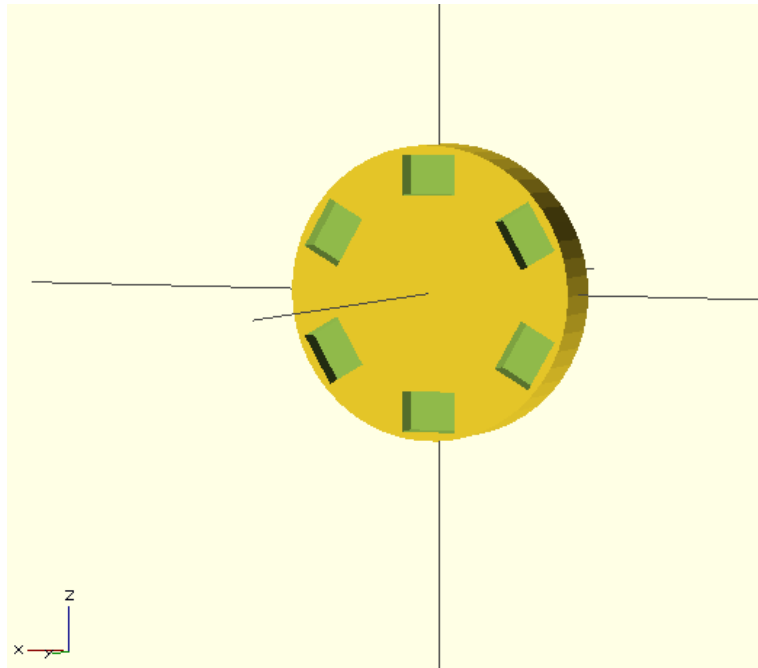


Figura 23 Cara trasera del puente de la muñeca

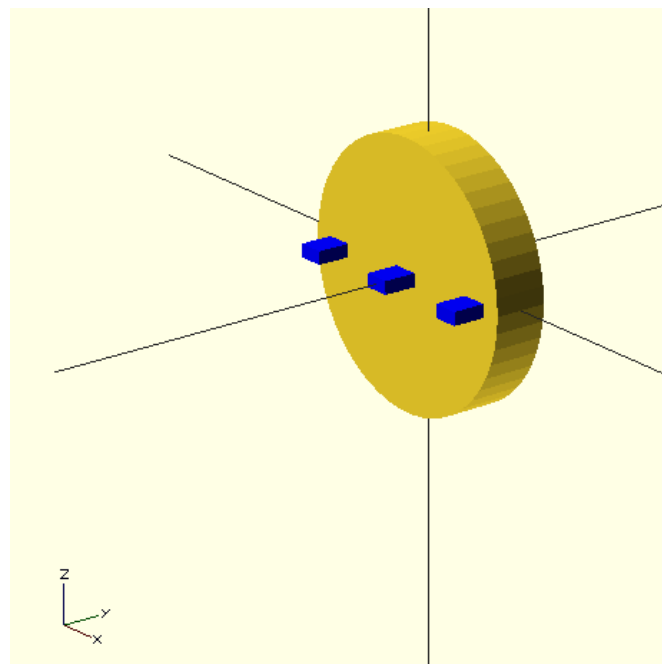


Figura 24 Cara frontal del puente de la muñeca

Resultado final de la muñeca

Finalmente, una vez se unen ambas piezas (bien por presión o bien con algún pegamento especial), el resultado final quedaría como sigue:

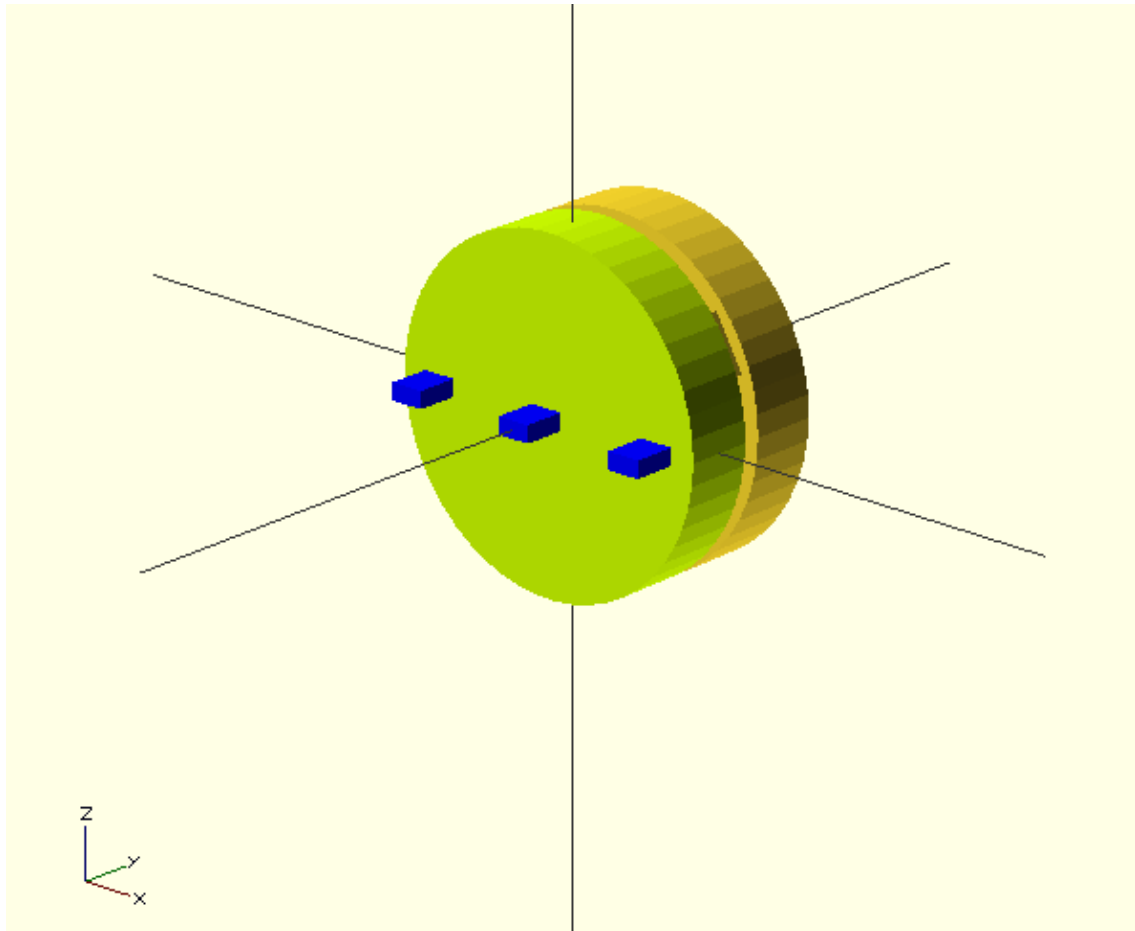


Figura 25 Resultado final de la muñeca

Es relevante comentar que una vez se hayan pegado estas dos piezas entre sí, no se podrá volver a colocar o retocar la corona del servomotor, por lo que hay que asegurarse de que esté colocada en la posición correcta para permitir el giro de 180° a partir de la posición deseada y no una aleatoria.

Por último, a partir de este módulo se incluirá la pinza, por lo que será necesario diseñar una superficie de contacto amplia en la base de la misma para que se pueda adherir con facilidad al puente de la muñeca. Es cierto que se han incorporado los salientes con esa idea, pero aún así es conveniente contar con más seguridad a la hora de juntar dos módulos.

3.2.6 Pinza

Base

La base es el elemento más crítico a la hora de diseñar una pinza, ya que las distancias han de estar calculadas para poder albergar los elementos necesarios. Hay cuatro aspectos críticos a la hora de diseñar una pinza de robot:

- Servomotor: Para que la pinza sea capaz de abrirse y cerrarse, necesita un servo que permita ese movimiento. Sin embargo, requiere del movimiento de dos piezas (los dos laterales del agarre), por lo que el movimiento del servo será transmitido a un par de engranajes que se ocuparán de este movimiento.
- Colocación de los engranajes: Los ejes de rotación de los engranajes han de estar en la misma línea. Uno de ellos, además, coincidirá con el eje del servomotor.
- Colocación de las barras de giro: Todas ellas tendrán la misma longitud y mantendrán la distancia entre las piezas de agarre. En base a su tamaño, se colocarán los agujeros para los tornillos de manera conveniente en la base.

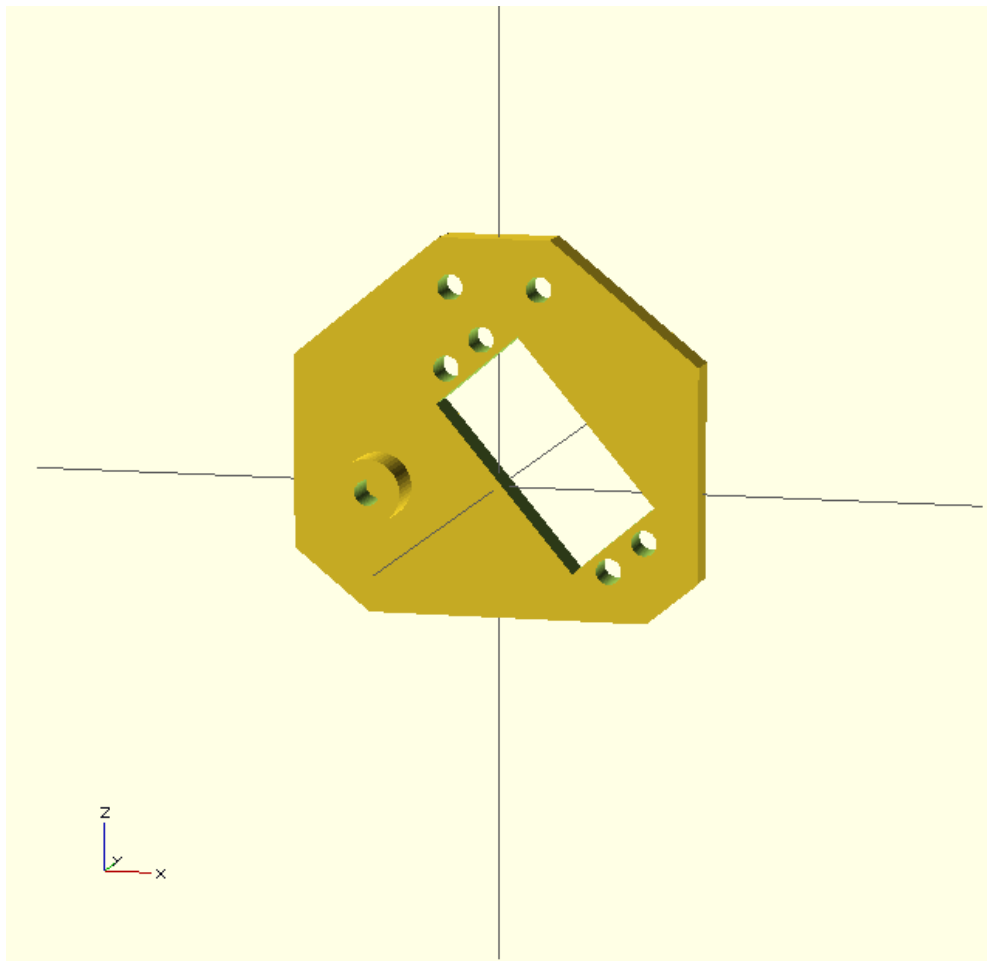


Figura 26 Base de la pinza

Engranaje

Los engranajes consistirán en unas ruedas dentadas que irán sujetas a la base de maneras diferentes. La primera albergará la corona del servomotor. La segunda, tendrá un agujero que encajará con el cilindro saliente de la base. De esta manera, mantendrá esta posición a la vez que engranará con su pieza simétrica.

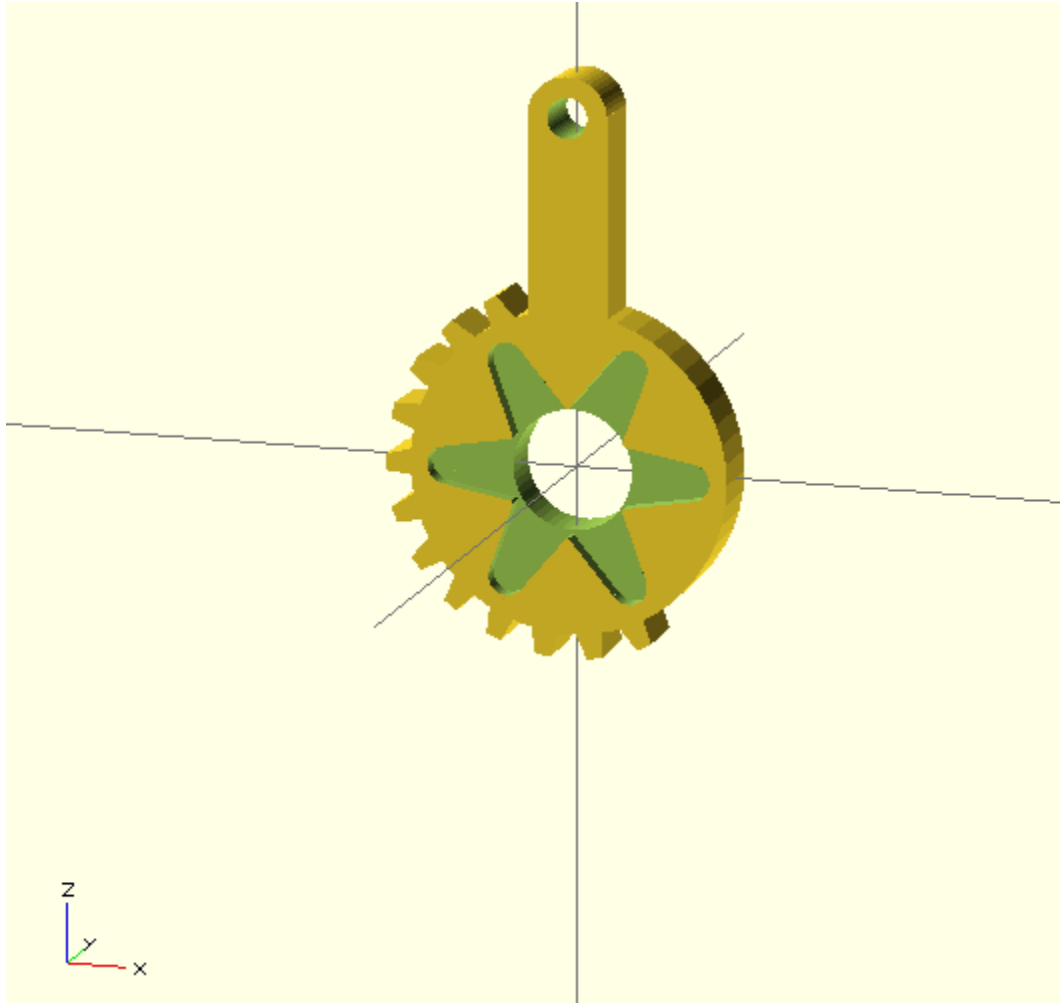


Figura 27 Engranaje de la pinza

El rectángulo redondeado que sobresale irá conectado a la pinza de agarre. El centro del agujero de esta pieza ha de corresponderse con el eje del servomotor. Además, los dientes han de encajar perfectamente con el otro engranaje. Como se puede observar, no es necesario diseñar toda la rueda dentada, pues la el movimiento de la pinza estará limitado (no es necesario una rotación de 180°).

Barras de giro

Las barras de giro son las piezas que conseguirán que el movimiento sea fijo en torno a ellas. Es decir, la rotación de los agarres estará condicionada por la longitud de esta pieza y rotará en torno al tornillo de sujeción entre la barra y la base:

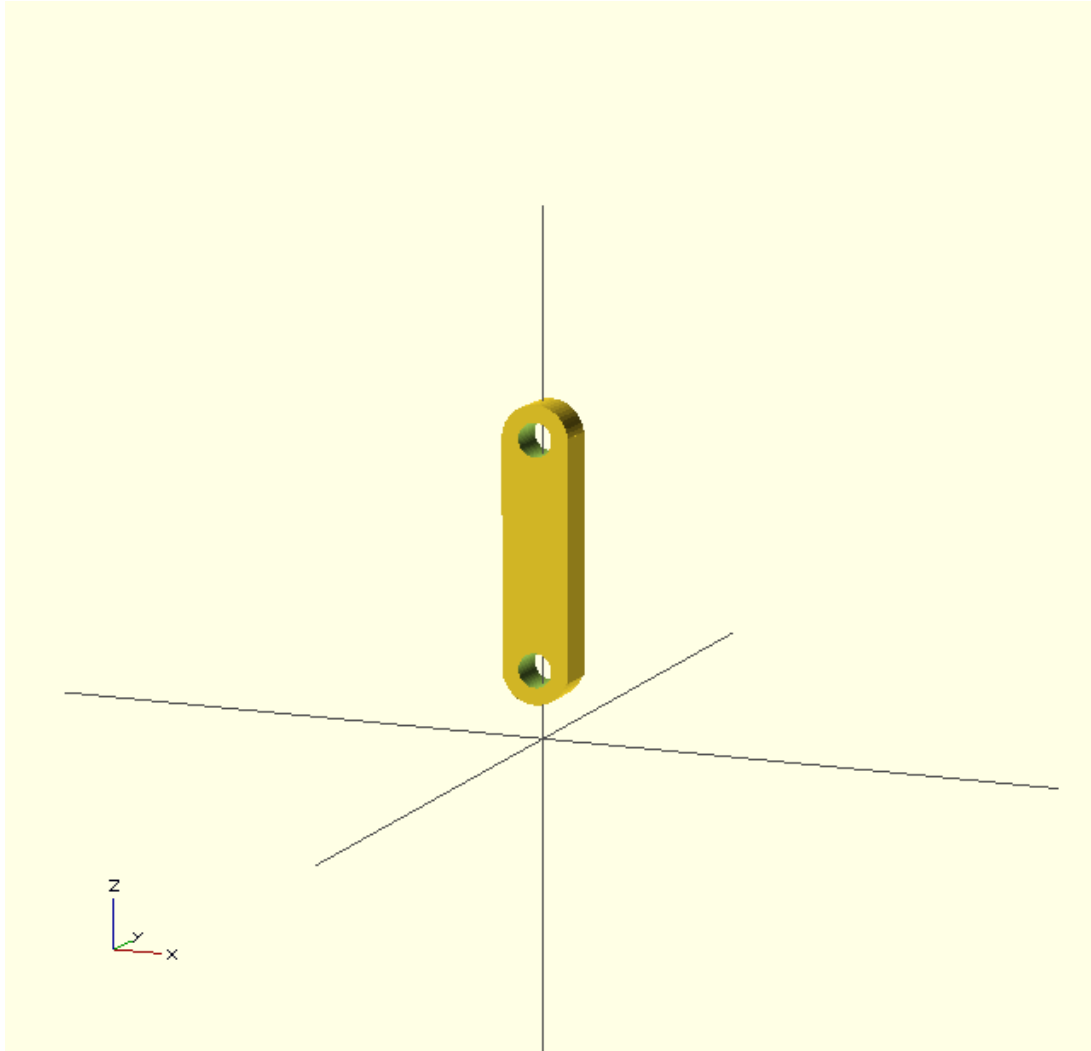


Figura 28 Barra de giro de la pinza

Se necesitarán 4 de estas piezas, ya que sujetaremos los agarres por encima y por debajo de la base. Estas barras estarán siempre de manera paralela a las barras salientes de los engranajes, consiguiendo así un movimiento restringido adecuado para un elemento aprehensor.

Agarre

Por último, la pieza de agarre es la más crítica, puesto que ha de tener un diseño tal que sea capaz de mantener piezas en suspensión. Por una parte, uno de sus tornillos unirá esta pieza con una barra de giro. Por otra parte, el segundo tornillo unirá esta pieza con el engranaje.

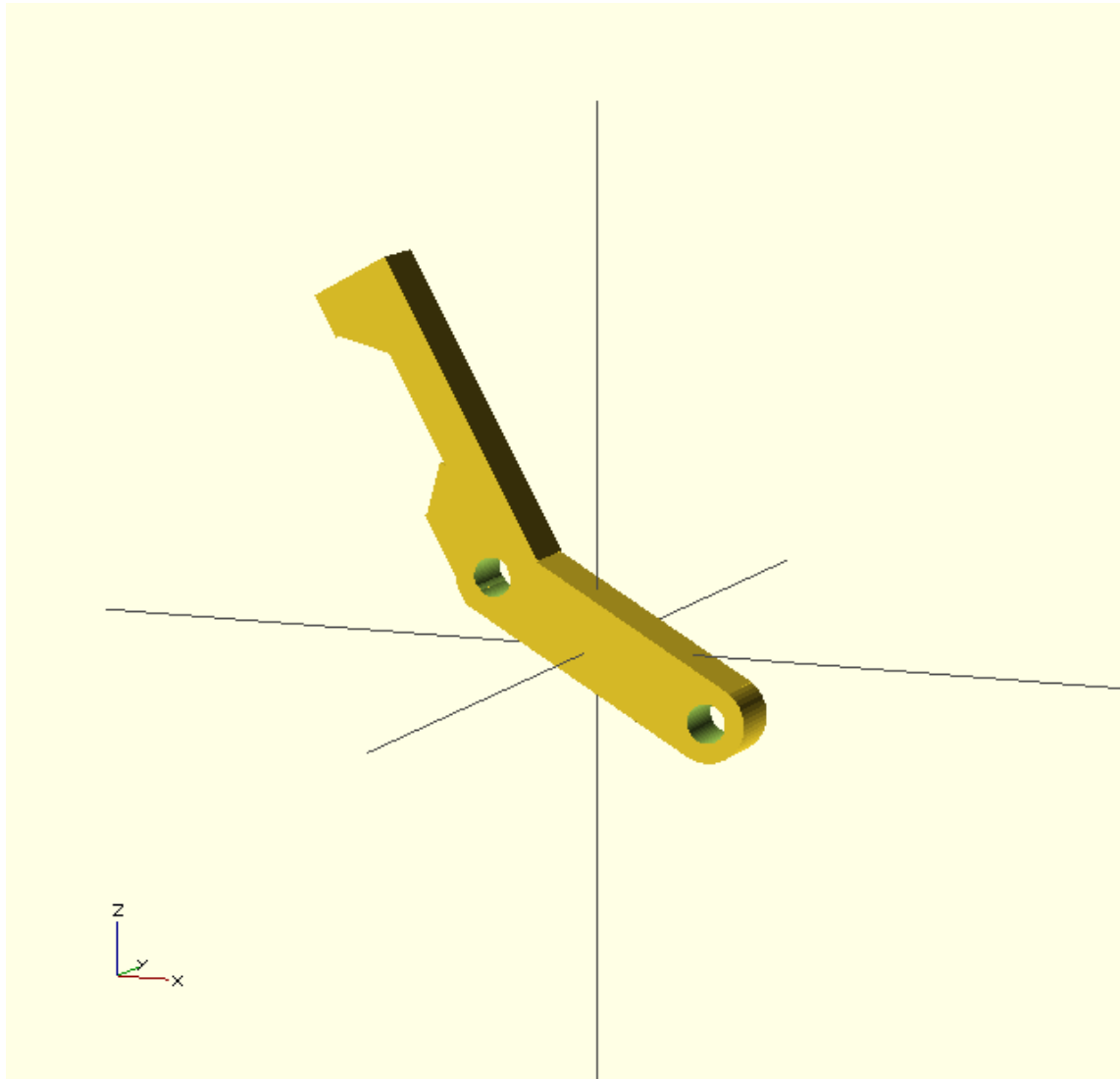


Figura 29 Agarre de la pinza

Hay que poner especial cuidado en el diseño de la parte inferior de esta pieza, pues no ha de chocar con las diagonales de la base de la pinza.

En este caso, el autor ha realizado una pequeña curvatura en la parte superior, para realizar una mayor sujeción entre la pieza y la pinza. Siempre se podría incluir alguna junta tórica para facilitar la adherencia entre pinza y objeto.

Resultado final

Finalmente, el último módulo de este brazo robótico se parecerá a la siguiente figura:

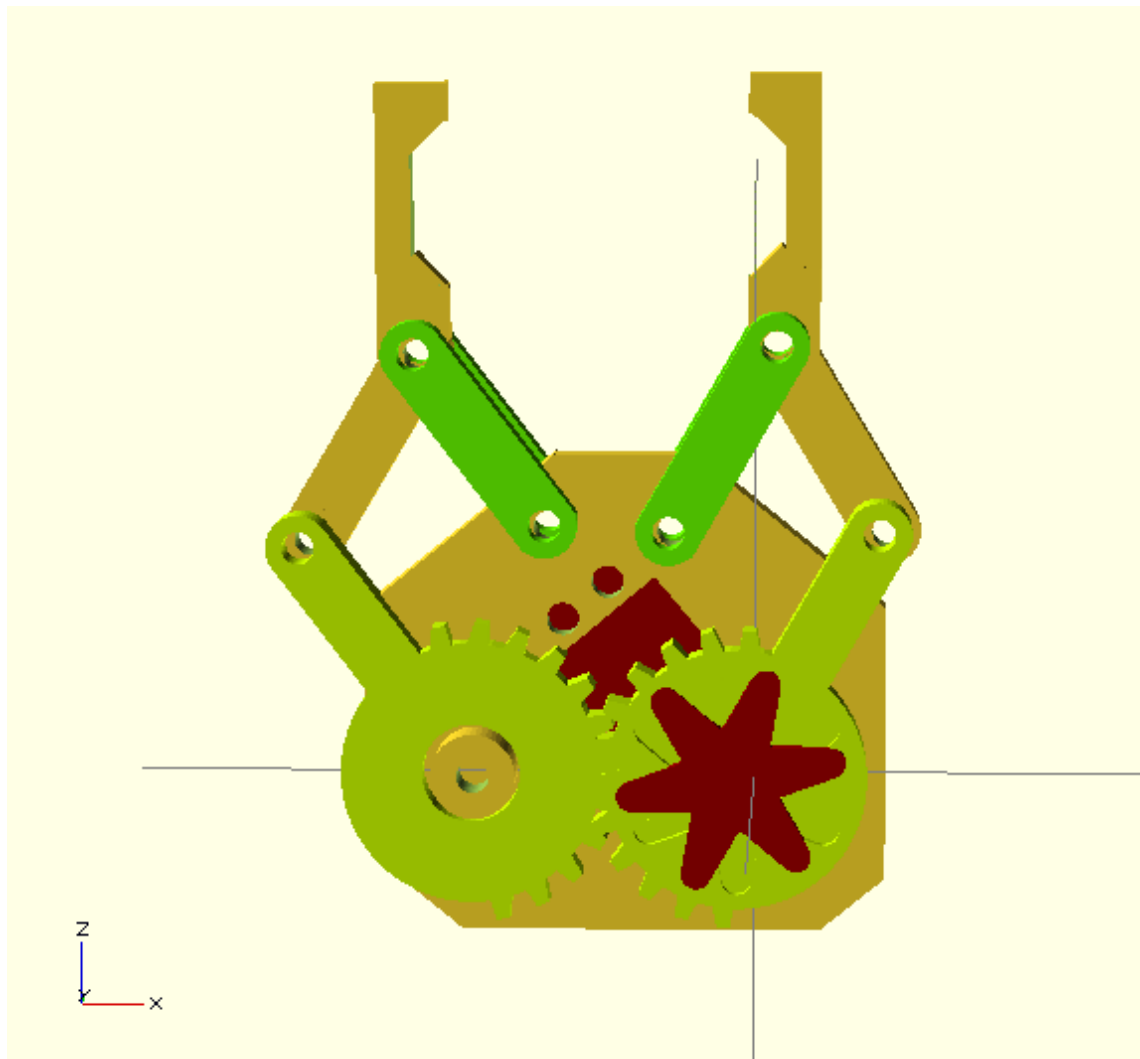


Figura 30 Resultado final de la pinza

Hay muchas maneras de diseñar una pinza y muchos modelos a elegir entre los diferentes útiles en un manipulador robótico. La elección de una pinza reside, sobre todo, en su simplicidad y su alta funcionalidad. Además, para otros modelos (como por ejemplo el de ventosa), se necesitarían elementos no imprimibles. Otros elementos terminales para este manipulador se podrían considerar mejoras del mismo.

3.2.7 Resultado final

El brazo completo tiene la siguiente forma:

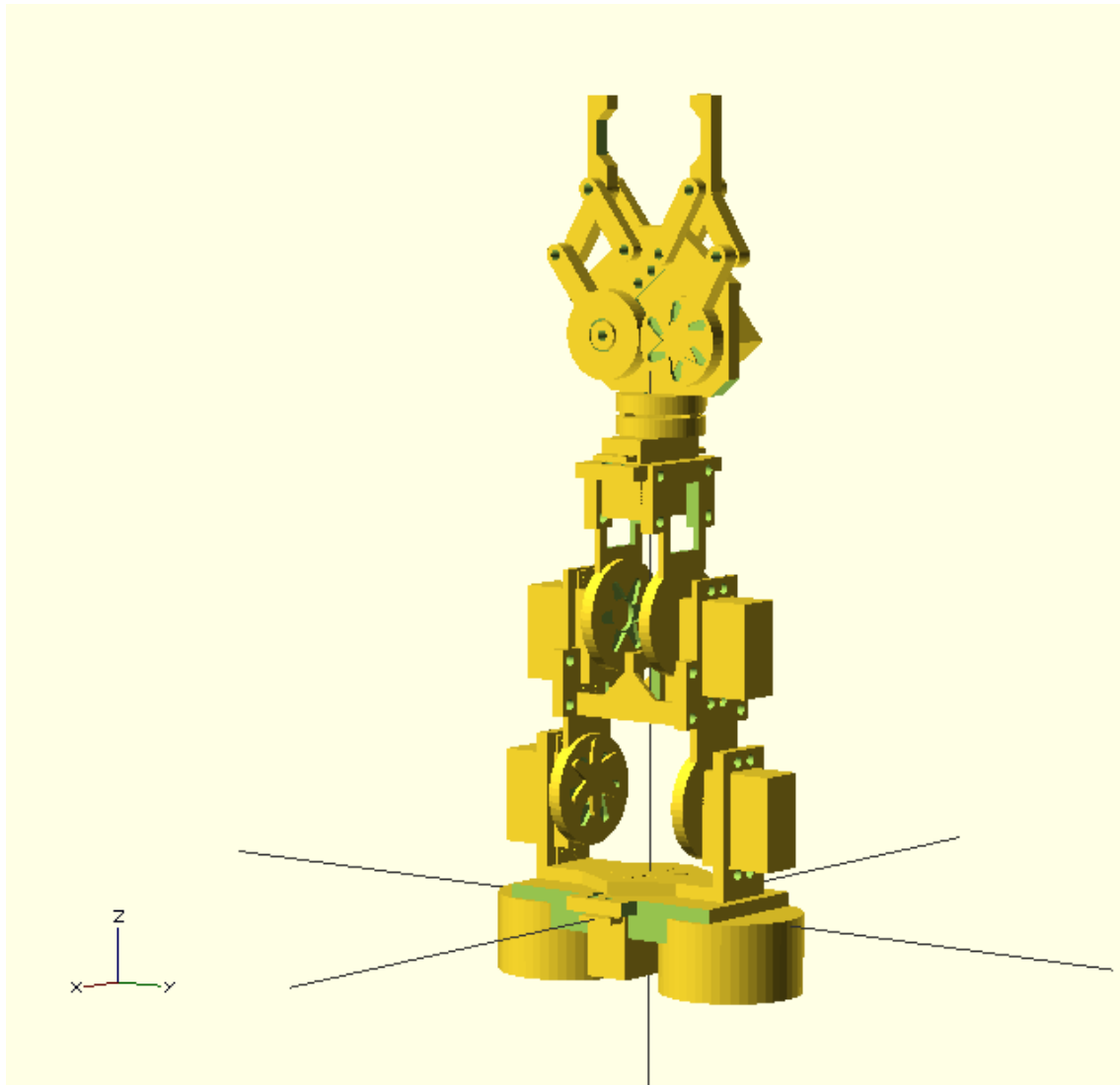


Figura 31 Resultado final del brazo completo

3.3 Cinemática directa

La cinemática de un robot estudia el movimiento del mismo con respecto a un sistema de referencia. Se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo y, en particular, por las relaciones entre la posición y la orientación de la herramienta del robot con los valores que toman las coordenadas de sus articulaciones.

La cinemática directa consiste en determinar la posición y orientación del extremo final del robot con respecto al sistema de la base del robot a partir del conocimiento de los valores de las articulaciones y sus parámetros geométricos. Se podría resumir de la siguiente manera:

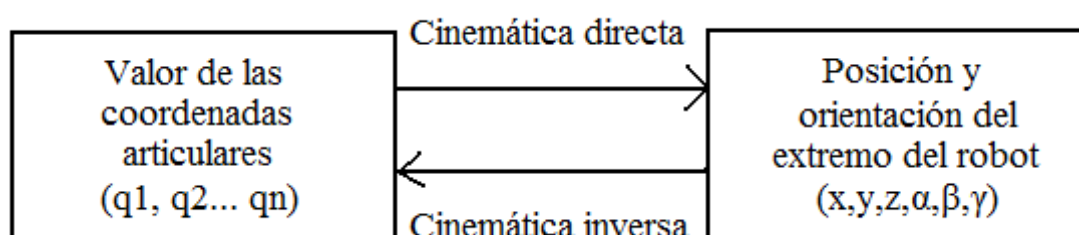


Figura 32 Cinemática directa e inversa

El problema cinemático directo se reduce a encontrar la matriz de transformación homogénea (A) que relaciona la posición y orientación del extremo del robot respecto a su sistema de referencia fijo (su base). Esta matriz A está en función de los parámetros de las articulaciones del robot. Para encontrar esta matriz, en 1955 Denavit y Hartenberg propusieron un método matricial que permite establecer de manera sistemática un sistema de coordenadas a través del cual es posible pasar de un eslabón a otro mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón:

- Rotación alrededor del eje Z_{i-1} un ángulo θ_i .
- Traslación a lo largo de Z_{i-1} una distancia d_i .
- Traslación a lo largo de X_i una distancia a_i .
- Rotación alrededor del eje X_i un ángulo α_i .

En definitiva, se dispondría de lo siguiente:

$$A_{i-1}^i = T(z, \theta_i) \cdot T(0,0,d_i) \cdot T(a_i, 0, 0) \cdot T(x,\alpha_i)$$

$$= \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cdot \cos\alpha_i & \sin\alpha_i \cdot \sin\theta_i & \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cdot \cos\alpha_i & -\sin\alpha_i \cdot \cos\theta_i & \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Así, encontrando esos parámetros, seremos capaces de determinar la cinemática directa del brazo robótico.

Imagen del robot en una posición inicial:

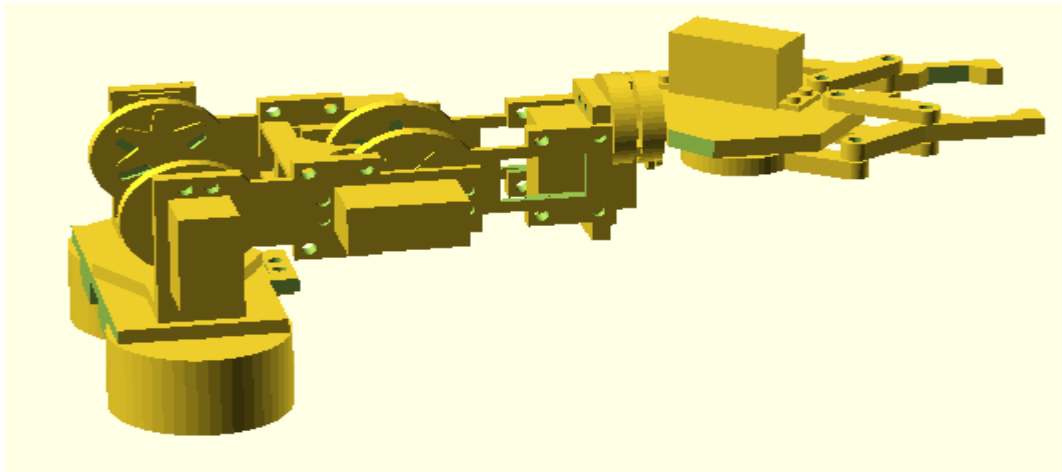


Figura 33 Cinemática del brazo robótico

Localización de las rotaciones en torno a los ejes:

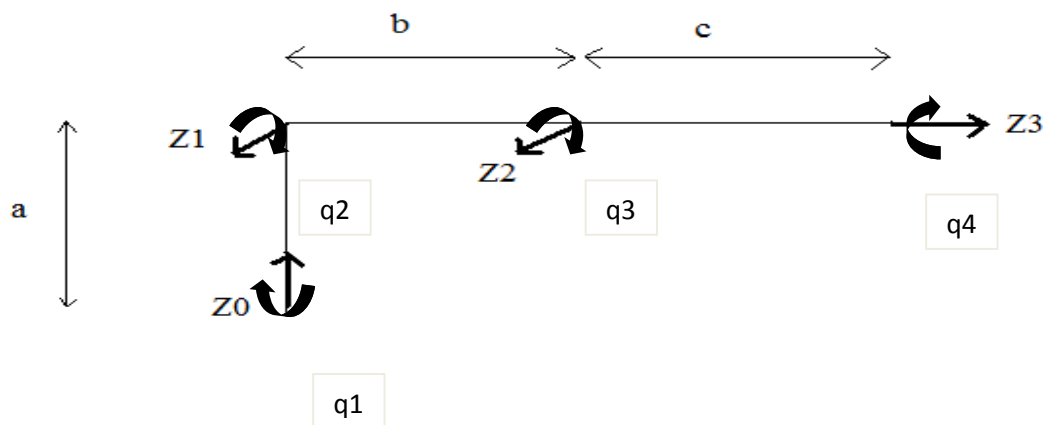


Figura 34 Ejes de rotación y geometría

Adjudicación de los sistemas de referencia dextrógiros:

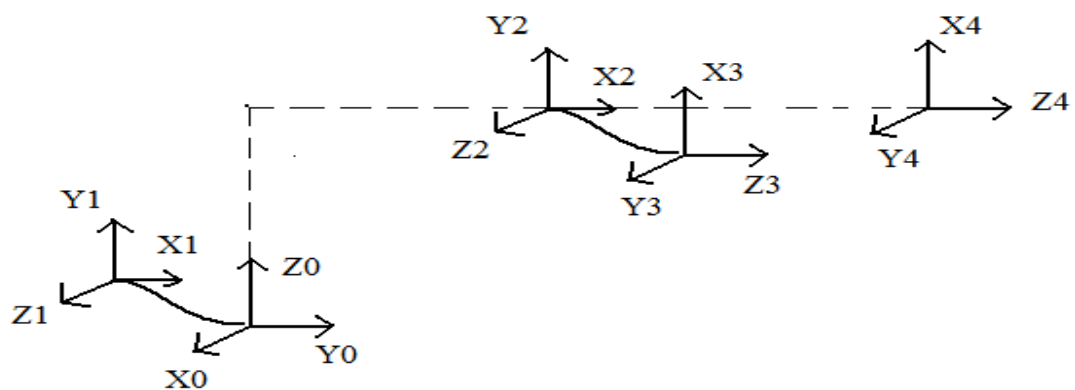


Figura 35 Sistemas de referencia finales

Parámetros de Denavit - Hartenberg:

i	θ_i	d_i	a_i	α_i
1	$\pi/2 + q_1$	0	0	$\pi/2$
2	$-q_2$	A	B	0
3	$-\pi/2 - q_3$	0	0	$\pi/2$
4	$-q_4$	C	0	0

Siendo $a = 45 \text{ cm}$;

$b = 75 \text{ cm}$;

$c = 80 \text{ cm}$;

CAPÍTULO 4:

PUESTA EN MARCHA

4.1 Electrónica

Los componentes necesarios son: 5 servomotores, placa arduino, clemas y una fuente de alimentación ATX.

Se necesita controlar y poner en marcha a 5 servomotores mediante la placa Arduino. Esta placa, como ya se ha comentado, dispone de todos los elementos necesarios para realizar el control de toda la electrónica. En este caso, utilizaremos los siguientes dispositivos:

- Conector USB (tipo A a tipo B):

Se utilizará para comunicar por puerto serie las órdenes de movimiento mediante el ordenador con sistema operativo Ubuntu. Además, suministrará 5V al microcontrolador para su funcionamiento.

- Pines digitales:

Estos pines reciben las señales PWM de posición de los motores desde el cable blanco de salida de los mismos. Se necesitarán por tanto 5 conexiones digitales asignadas de la siguiente manera:

Pines digitales	Control
Pin 3	Base
Pin 6	Antebrazo
Pin 9	Brazo
Pin 10	Muñeca
Pin 11	Pinza

Nota: No es del todo necesario utilizar una señal de control PWM para el abrir-cerrar de la pinza, puesto que se puede simplemente asignar un ángulo máximo de cierre y de apertura para facilitar el control de la misma. Aún así, se podrá disponer de un pin digital para esta funcionalidad.

- Pines analógicos:

Al trazar los servos, hemos obtenido 5 nuevos cables que se han de conectar a los pines analógicos de la placa. Éstos transmitirán la información del potenciómetro interno del motor al software de Arduino, desde el cual se leerá la posición y se actuará en consecuencia. Se asignarán estos pines de la siguiente forma:

Pines analógicos	Posición
Pin A4	Base
Pin A2	Antebrazo
Pin A3	Brazo
Pin A1	Muñeca
Pin A0	Pinza

Ya se han determinado todas las conexiones que han de realizarse respecto al control y la comunicación entre servomotores, la placa y el ordenador. Ahora se han de explicar las conexiones entre la fuente de alimentación y el resto de dispositivos para proporcionar el voltaje y amperaje necesarios para su funcionamiento:

- Puentear cable verde con cable negro en la fuente ATX:

Esto permite el funcionamiento de la fuente de manera independiente ya que conecta el "power on" de la fuente de alimentación (cable verde) con la tierra (cable negro). De esta manera, al encender la fuente, se activará el ventilador, señal de que se está proporcionando la tensión necesaria.

- Referenciar la tierra de la placa Arduino con la tierra de la fuente ATX:

Recordemos que al trucar los servomotores, únicamente hemos soldado un cable auxiliar para obtener la posición. Esto significa que las tierras de los servomotores han de ir conectadas directamente a la fuente. Paralelamente, hay que darse cuenta de que no estamos referenciando la tierra de la placa Arduino a ningún sitio, por lo que no funcionaría si no conectáramos el pin GND de la placa con el cable negro de la fuente.

- Unir todos los cables negros y GND de Arduino con la tierra de la fuente ATX:

Separando los diferentes cables negros de todos los servos, los pelamos por el extremo y utilizamos una clema para obtener la misma conexión con un solo cable. Éste irá conectado al cable negro de la fuente. No podemos olvidarnos de la referencia de tierra de la Arduino.

- Unir todos los cables rojos con la alimentación de la fuente ATX:

En la fuente de alimentación se proporcionan varios cables con diferentes tensiones de alimentación y amperajes. En este caso, la tensión admitida por los servomotores es de 4 a 6V, por lo que se utilizará el cable rojo (5V, 25A).

- Opcional: Alimentación externa de Arduino:

Ya se ha hablado de la posibilidad de alimentar de manera externa la placa en lugar de utilizar el cable USB a través del cual también se transmite la comunicación. Si quisiéramos realizar este paso, se conectaría el pin VIN con el cable amarillo de la fuente (12V, 0.8A).

En resumen, toda la electrónica se comunica con el PC mediante la placa Arduino, alimentada mediante la conexión USB. La alimentación de servos se realiza de manera externa (fuente ATX), para que no se sobrecaliente la electrónica.

4.1.1 Diagrama de conexiones

En la siguiente figura , se ilustran las conexiones necesarias entre la arduino y los servomotores:

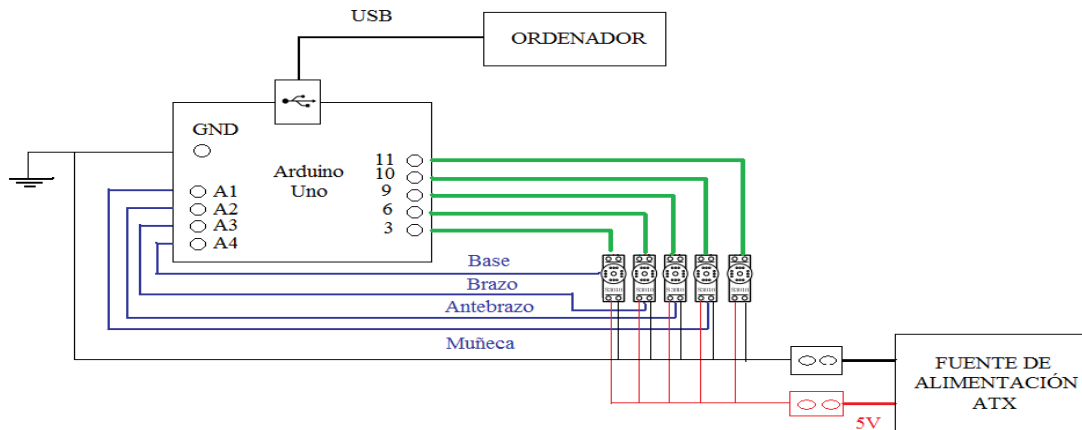


Figura 36 Diagrama de conexiones

4.1.2 Elementos necesarios

Ahora que ya se han definido todas las conexiones, es el momento de montar el robot para poder unir todos los cables entre sí y poder pasar a la parte del control de los módulos. Los elementos de los que dispondremos finalmente para montar el robot serán, por tanto, los siguientes:



Figura 37 Elementos de montaje del robot

4.2 Alimentación

Se ha hablado de cómo la placa de Arduino está conectada mediante USB al ordenador para realizar la comunicación por puerto serie, pero mediante este cable también se puede alimentar la placa. Así, se podrían conseguir los 5V necesarios para la alimentación de cada servo. Sin embargo, esta opción no es válida en el momento en el que tienes más de un motor, puesto que el puerto USB limita la intensidad a 500 mA y este valor no es suficiente para repartir entre todos los motores. Por tanto, el autor ha optado por utilizar una fuente de alimentación ATX que ha extraído de una vieja CPU de ordenador.



Figura 38 Fuente de alimentación ATX

Esta fuente lleva conectados varios cables, pero solo nos interesan los más esenciales: el cable de tierra y el de alimentación a 5V. Se puede comprobar en las características técnicas que estos 5V proporcionan 25A, lo que será más que suficiente para mover los módulos del robot.

La placa de arduino será alimentada mediante USB, puesto que no requiere más voltaje del proporcionado por este medio, aunque sería posible conectar el cable amarillo de la fuente (12V, dentro del rango de voltaje de entrada) al conector de Vin de la placa si quisiéramos mantener apartada toda la electrónica de la alimentación, evitando así sobrecalentamientos innecesarios.

4.3 Control

4.3.1 Aspectos a controlar

La idea para controlar el brazo consiste en la pulsación de una tecla que indique la cantidad de grados que el módulo debe moverse. Por ejemplo, podemos asignar una cantidad fija de 10 grados cada vez que se pulse una tecla. De esta manera, si pulsáramos "W", se estaría indicando que el usuario quiere que el módulo se mueva de manera ascendente 10 grados respecto de la posición en la que se encuentra en ese momento.

Al iniciar el software, el brazo ha de estar en reposo (completamente en vertical). Teniendo en cuenta el área de trabajo del robot, esto significa que los módulos han de estar en las siguientes posiciones:

Módulos	Posición
Base	0º
Antebrazo	180º
Brazo	105º
Muñeca	160º
Pinza	Máximo ángulo de apertura

Posteriormente se realiza la elección del módulo a mover mediante las teclas de acceso rápido. Finalmente, se aplica la rotación deseada mediante incrementos de 10 grados por pulsación. Estos funcionamientos quedan reflejados en los siguientes diagramas de flujo:

Elección de la posición de reposo:

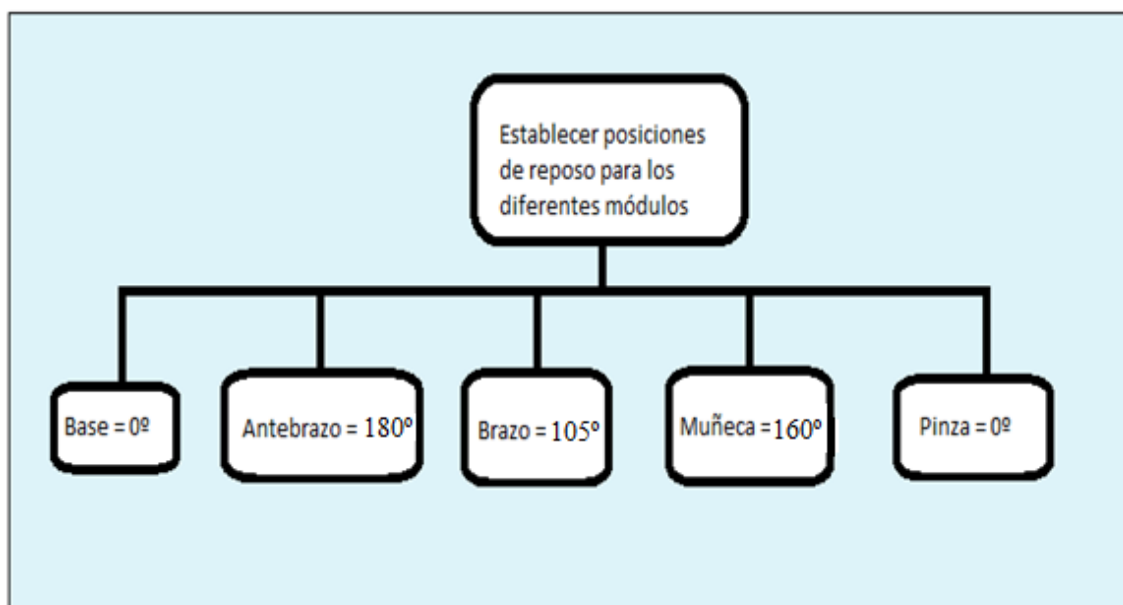


Figura 39 Flujograma 1

Elección del módulo a mover:

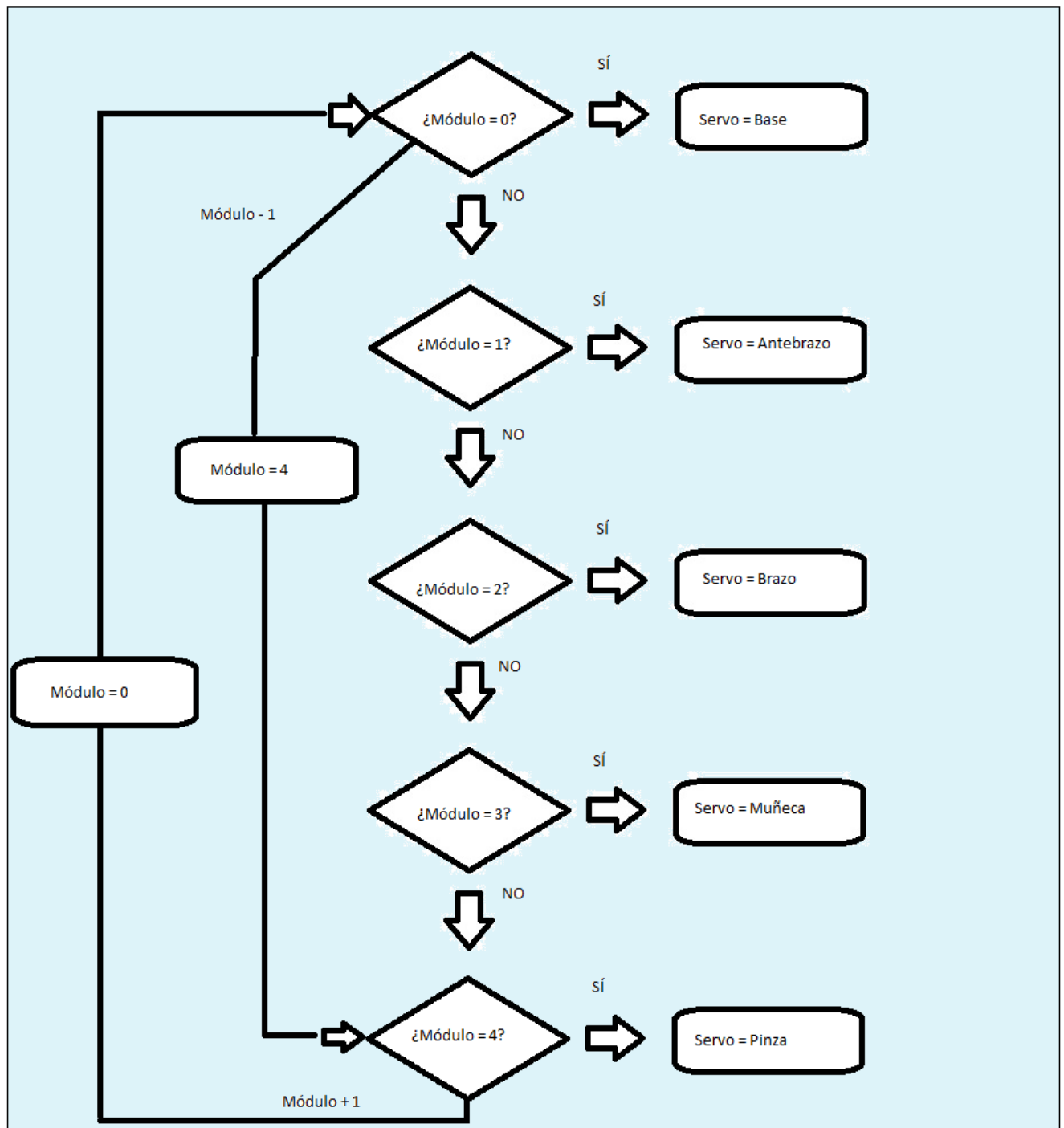


Figura 40 Flujograma 2

Además de asignar un módulo a cada valor de la variable, se ha implementado una solución recurrente, de manera que si se aumenta el módulo a partir del último (la pinza), se seleccionará la base y viceversa en el caso decreciente.

De manera análoga:

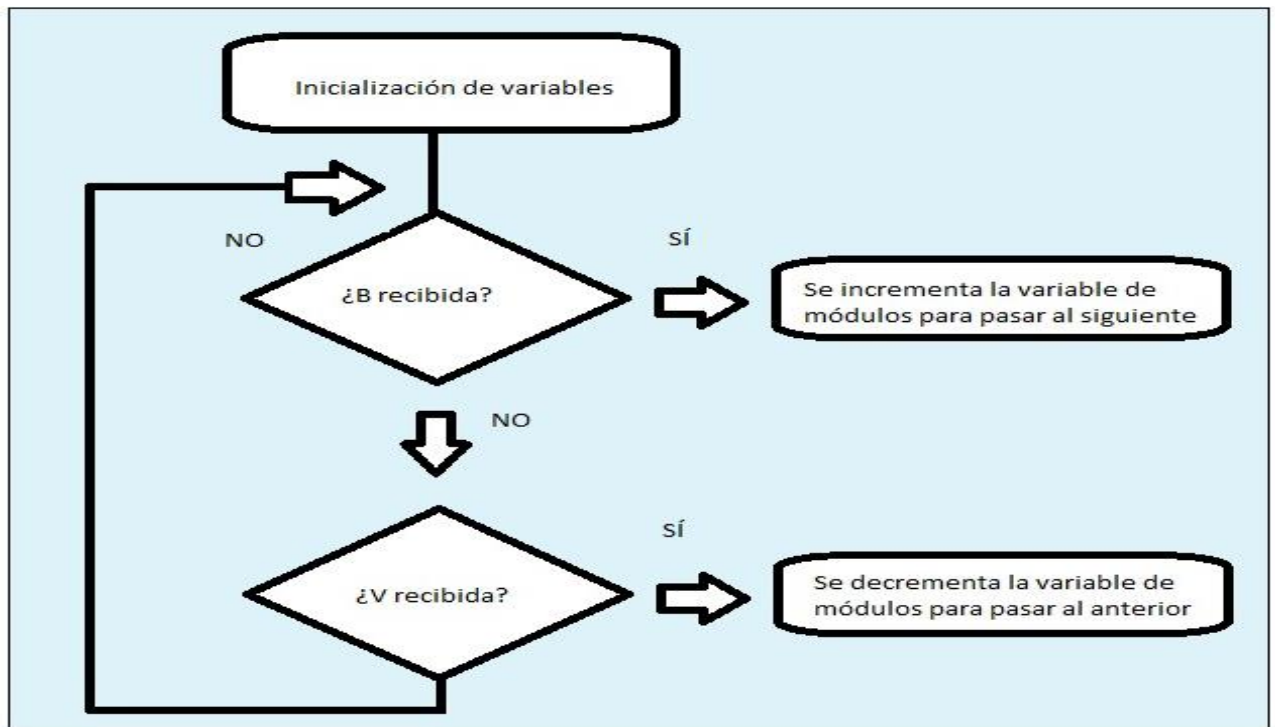


Figura 41 Flujograma 3

Realización del movimiento:

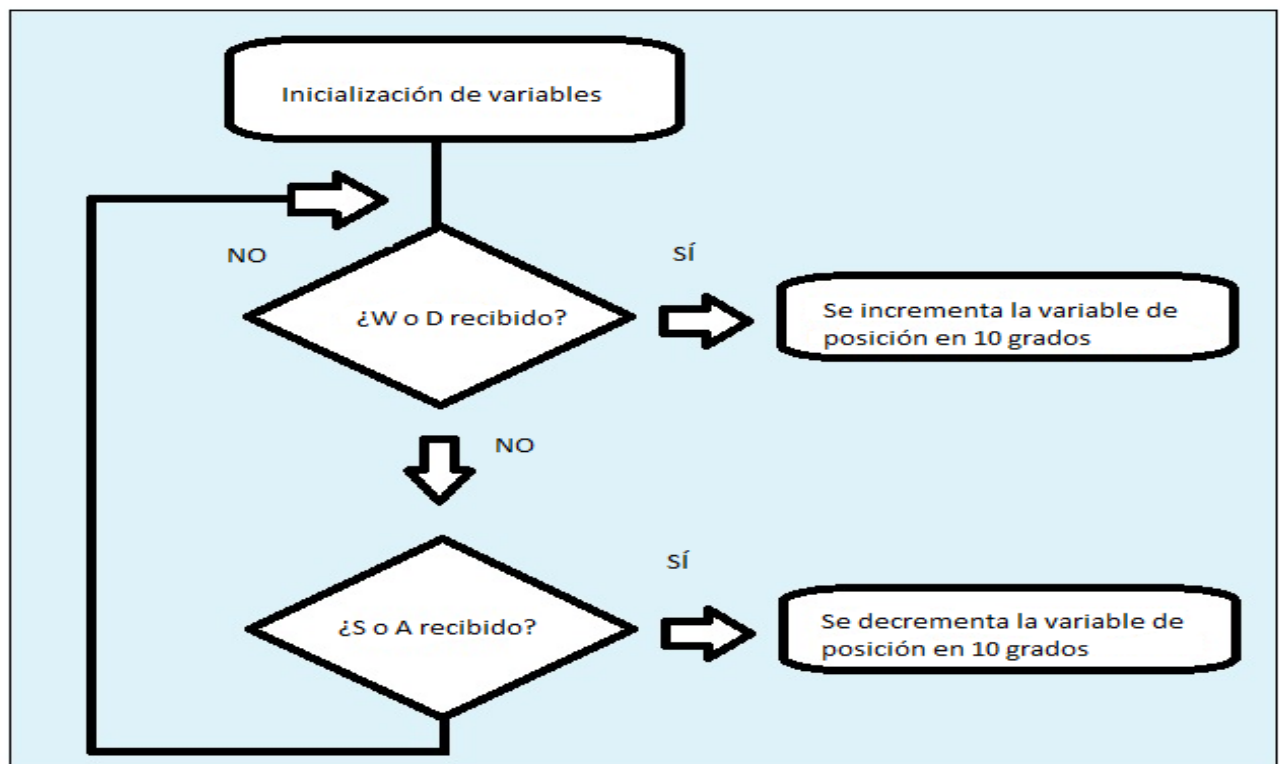


Figura 42 Flujograma 4

Por último, se escribe el valor de la posición en el servomotor correspondiente:

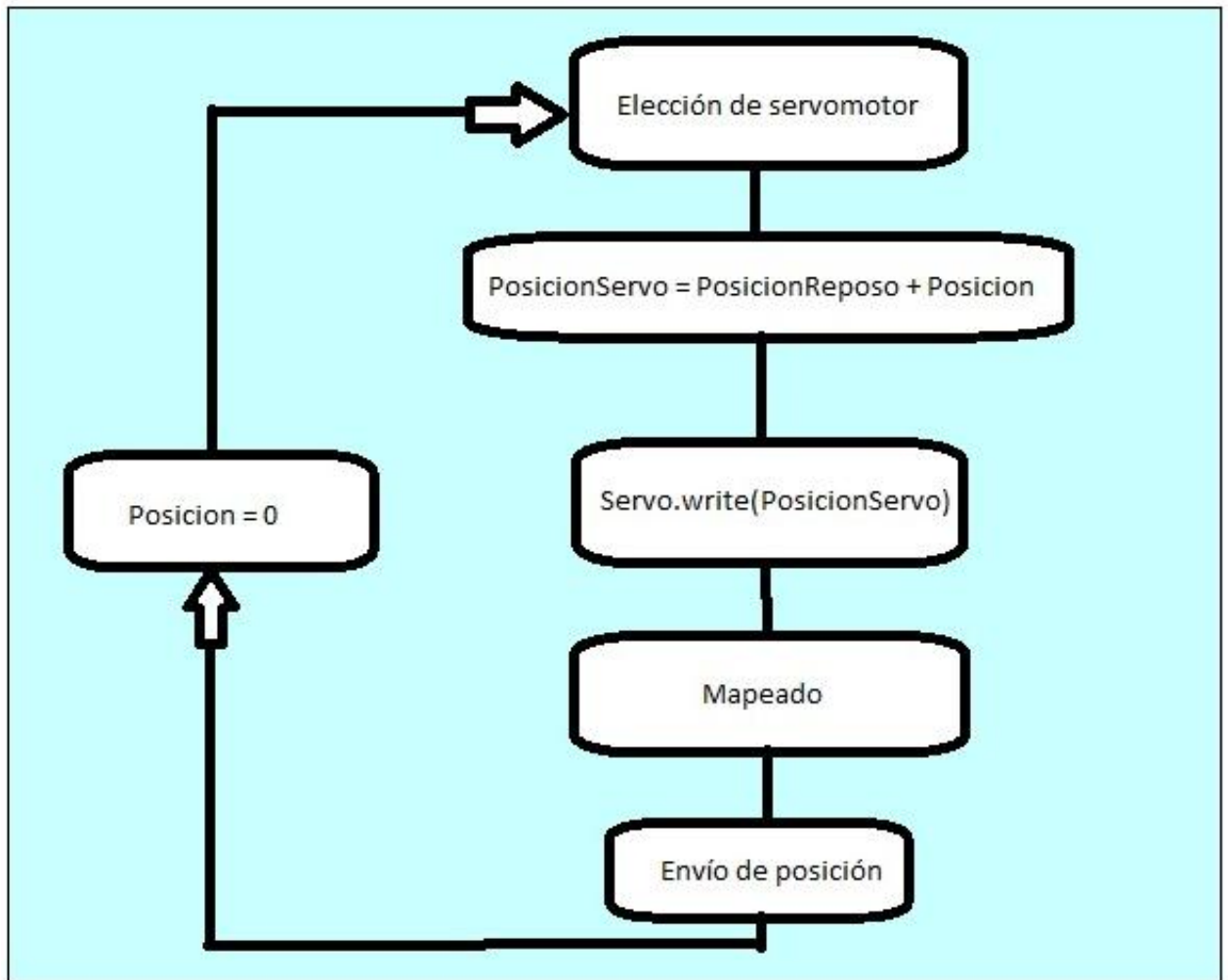


Figura 43 Flujograma 5

4.3.2 Mapeado

El envío de la posición actual mediante la sentencia "analog.read()" y que se recibe mediante el cable soldado no es una parte directa del control del brazo, sino que está proyectada hacia futuras modificaciones y mejoras en el control del robot.

Ya se ha explicado que este cable, soldado a la placa a la señal del potenciómetro, es la que nos permitirá obtener la posición real del motor. Sin embargo, debido a la soldadura, es posible (y a demás, muy previsible) que haya "ruido" en la señal. Por ello, es necesario filtrarla, o bien de manera física (añadiendo un condensador o cualquier otro elemento capacitor en una placa protoboard) o bien mediante el software. En este caso, el autor ha decidido aplicar un mapeado de la posición en el software de Arduino, de manera que se ajusta esta señal de manera experimental.

La señal del potenciómetro oscila entre el valor 0 y el valor 1023 (10 bits), por lo que se realiza en primer lugar un mapeado entre estos valores y 0 y 180 grados (para que se convierta este valor en un valor de posición). Se obtienen los siguientes resultados, considerando 3 secuencias diferentes:

PosiciónEnviada\PosiciónRecibida	Primera secuencia	Segunda secuencia	Tercera secuencia
0	23	23	23
10	27	26	26
20	31	30	30
30	35	35	35
40	39	39	39
50	43	43	43
60	46	46	46
70	50	50	50
80	54	54	54
90	57	57	57

Como se puede observar, los valores distan mucho de ser los valores reales, así que se realizarán mapeados auxiliares para ayudar a aproximar este valor al verdadero. En primer lugar, se realiza un mapeado entre 0 y 5000 (voltaje mínimo y máximo) para obtener una relación mayoritariamente lineal y luego poder volver a mapear a la posición en grados. Se obtienen los siguientes resultados:

PosiciónEnviada\VoltajeAplicado	Primera secuencia	Segunda secuencia	Tercera secuencia
0	645	669	669
10	757	762	767
20	884	874	884
30	997	977	992
40	1109	1099	1109
50	1217	1212	1212
60	1319	1319	1314
70	1422	1417	1417
80	1520	1520	15151
90	1617	1612	1612
100	1715	1710	1700
110	1813	1813	1808
120	1901	1911	1906
130	1994	1999	1994
140	2091	2091	2086
150	2179	2179	2174
160	2272	2267	2267
170	2399	2394	2394

Finalmente, se utilizan estos valores (De 645 a 2400) para realizar de nuevo el mapeado a posición, obteniendo lo siguiente:

PosiciónEnviada\PosiciónRecibida	Primera secuencia	Segunda secuencia	Tercera secuencia
0	0	0	0
10	9	9	10
20	20	20	21
30	33	33	32
40	44	44	43
50	55	53	55
60	65	65	66
70	74	74	73
80	84	84	84
90	92	90	92
100	103	102	103
110	113	114	113
120	121	121	121
130	130	131	131
140	140	142	142
150	152	152	153
160	162	161	160
170	170	170	170

Ahora ya sí se pueden considerar unos valores certeros (con un nivel de error de ± 5 grados, pero minimizado respecto a lo inicial).

4.4 Aspecto real del robot

Una vez se han ensamblado los diferentes módulos, el aspecto del brazo queda definido en la siguiente figura:

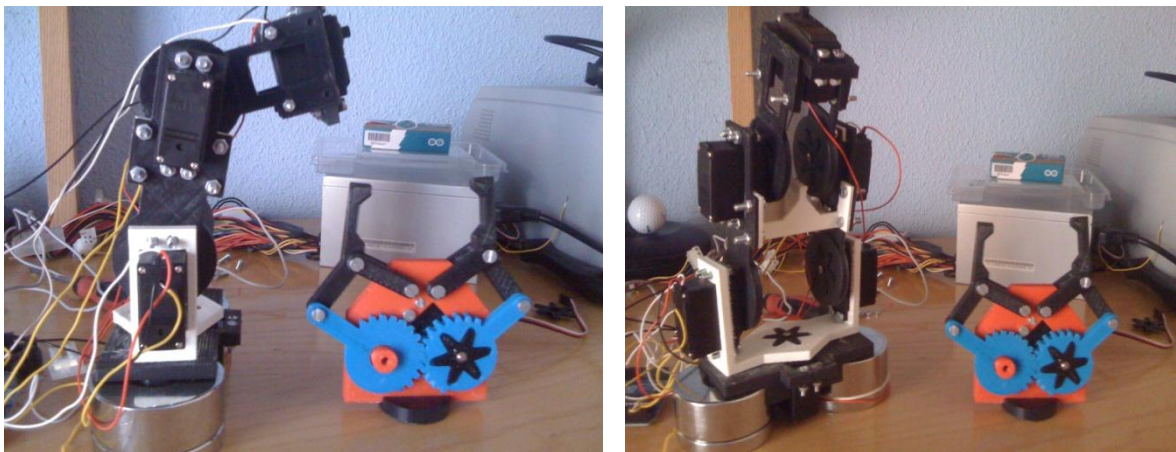


Figura 44 Aspecto real del brazo

Una vez se ha colocado la pinza, el brazo final tiene el siguiente aspecto:

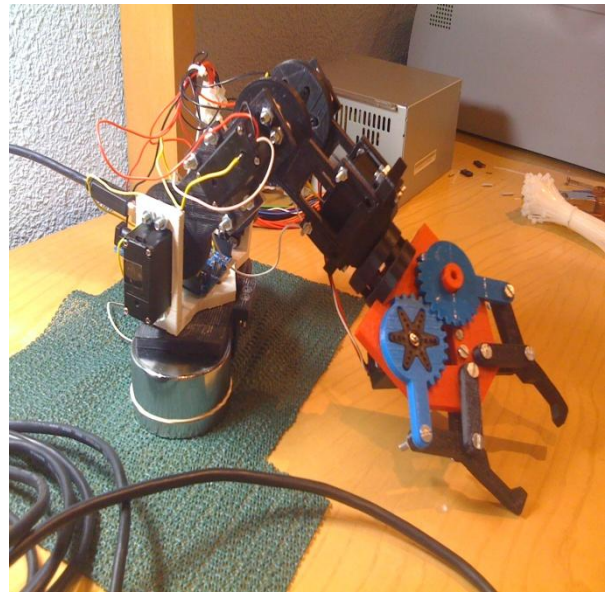
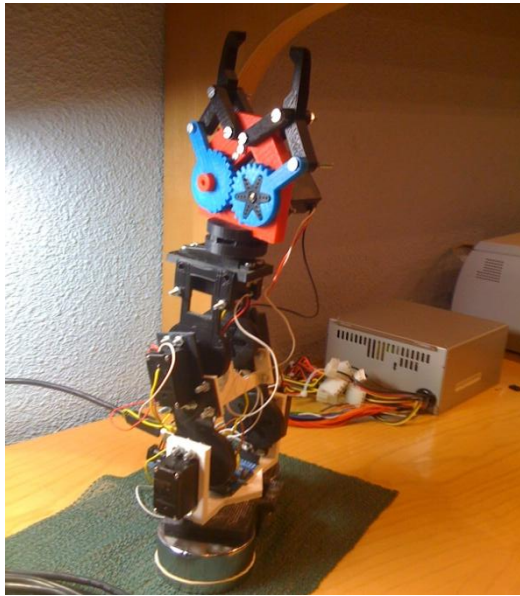


Figura 45 Brazo con pinza

Por último, se puede comprobar el funcionamiento del brazo en el siguiente enlace, donde se ejecutan los siguientes movimientos:

- Colocación de la pinza en la posición de agarre.
- Giro del antebrazo a una posición más baja.
- Giro del brazo a la posición del objeto.
- Se cierra la pinza.
- Se levanta el brazo y se gira la base para llevar el objeto a otra posición.
- Se sitúa el brazo en la posición donde se ha de dejar el objeto.
- Se abre la pinza para depositar el objeto.
- Vuelta a la posición de reposo.

Enlace del vídeo: <http://youtu.be/sUzbNI0DV4Y>

Adicionalmente, se puede observar el funcionamiento con control a tiempo real (mediante el teclado) en el siguiente vídeo:

<http://youtu.be/sbEiRr2WQlk>

4.5 Conclusiones

Al comienzo de esta memoria se establecieron unos objetivos y subobjetivos que se han conseguido cumplir. En primer lugar, se ha conseguido diseñar todos los módulos necesarios para el funcionamiento del brazo permitiendo la posibilidad de inclusión de más módulos con diferentes diseños. Esto significa que no solo se ha llegado a una solución donde el brazo es funcional, sino que además se pueden incluir más módulos o incluso modificaciones de los anteriores de una forma sencilla. Esto favorece el diseño y la creación, puesto que otra persona puede empezar a trabajar desde el punto en el que el autor ha terminado este proyecto, valorando el intercambio de ideas y la compartición de las mismas.

Este proyecto, adicionalmente, ha sido incluido en la página web de Thingiverse, desde donde se irán incluyendo modificaciones y se puede ayudar a nuevos usuarios a replicar el robot.

Finalmente, se ha implementado una solución para el control del brazo y se han asentado las bases para conseguir un control mediante teclas de acceso rápido al teclado.

4.6 Mejoras y trabajos futuros

A lo largo de la realización de este proyecto, han surgido diferentes inconvenientes que se podrían solventar llevando a cabo ciertas mejoras del modelo propuesto en esta memoria. Estas exceden por muchos motivos los objetivos principales que se establecieron al iniciar este proyecto, por lo que las englobo en el marco de trabajos futuros. Podemos destacar las siguientes:

Diseño de nuevas pinzas:

La pinza utilizada en este proyecto contiene todos los elementos para su funcionamiento, pero no está optimizada. Se puede reducir el tamaño y mejorar las piezas de agarre.

Diseño de nuevos elementos aprehensores:

¿Por qué conformarse con la tradicional pinza?. Hay muchos y muy diversos elementos que permiten manipular un objeto: una ventosa, un imán... El hecho de que la pinza sea un módulo independiente permite cambiar de diseño simplemente desatornillando la corona del último módulo del brazo.

Inclusión de más grados de libertad:

Una vez se ha demostrado la viabilidad del brazo mediante su funcionamiento, se pueden incluir más módulos que permitan más giros y más posiciones alcanzables. Por ejemplo, se podría incluir un grado de libertad más en la muñeca para que no solo permita la rotación en Z, sino también en el eje Y. También hay que remarcar el hecho de que el autor ha decidido usar unas pesas para sujetar el brazo, pero se podrían adherir unos carriles que, además de permitir el movimiento de manera lineal, sujete de manera permanente el brazo.

Servomotores de brazo y antebrazo sincronizados:

Se ha hablado del método de unión de los módulos mediante la corona estrellada y de la utilización de un motor vacío que permita el giro libre de su simétrico, pero se puede realizar un control exhaustivo de la posición de ambos servos para conseguir pares mayores que permitan manipular objetos de mayor tamaño.

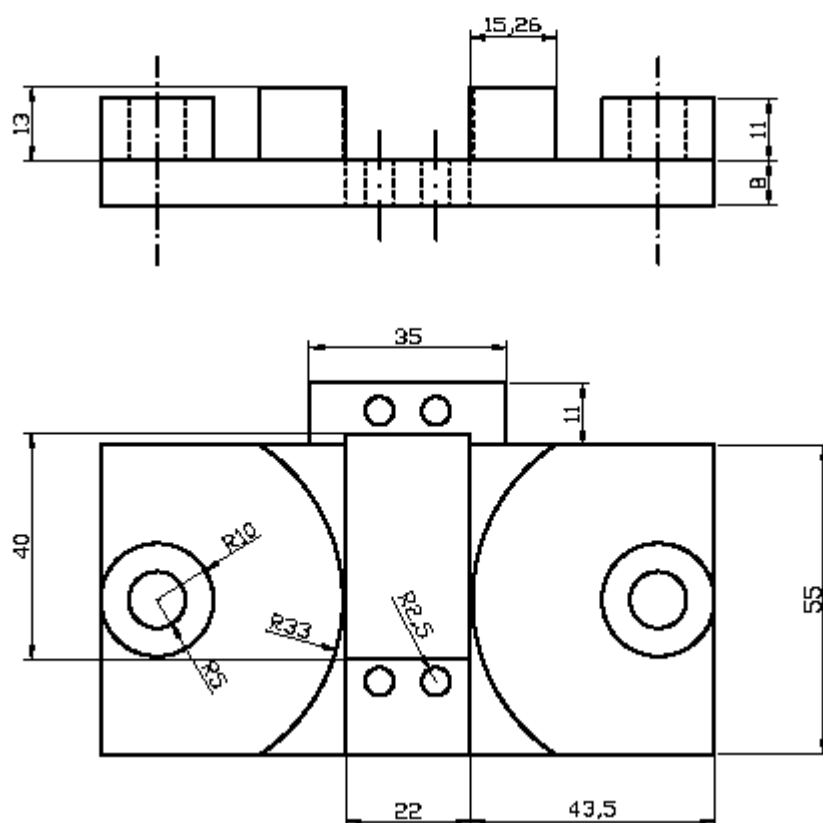
Control por bluetooth:

La comunicación por puerto serie permite la adición de un dispositivo bluetooth donde no sea necesario un cable USB para transmitir los datos de posición del robot. Para ello habrá que alimentar arduino desde una fuente externa, pero no habría inconveniente puesto que la solución implementada por el autor ya preveía mejoras de este tipo y ya se ha incluido una fuente de alimentación externa ATX que posibilita este tipo de cambios.

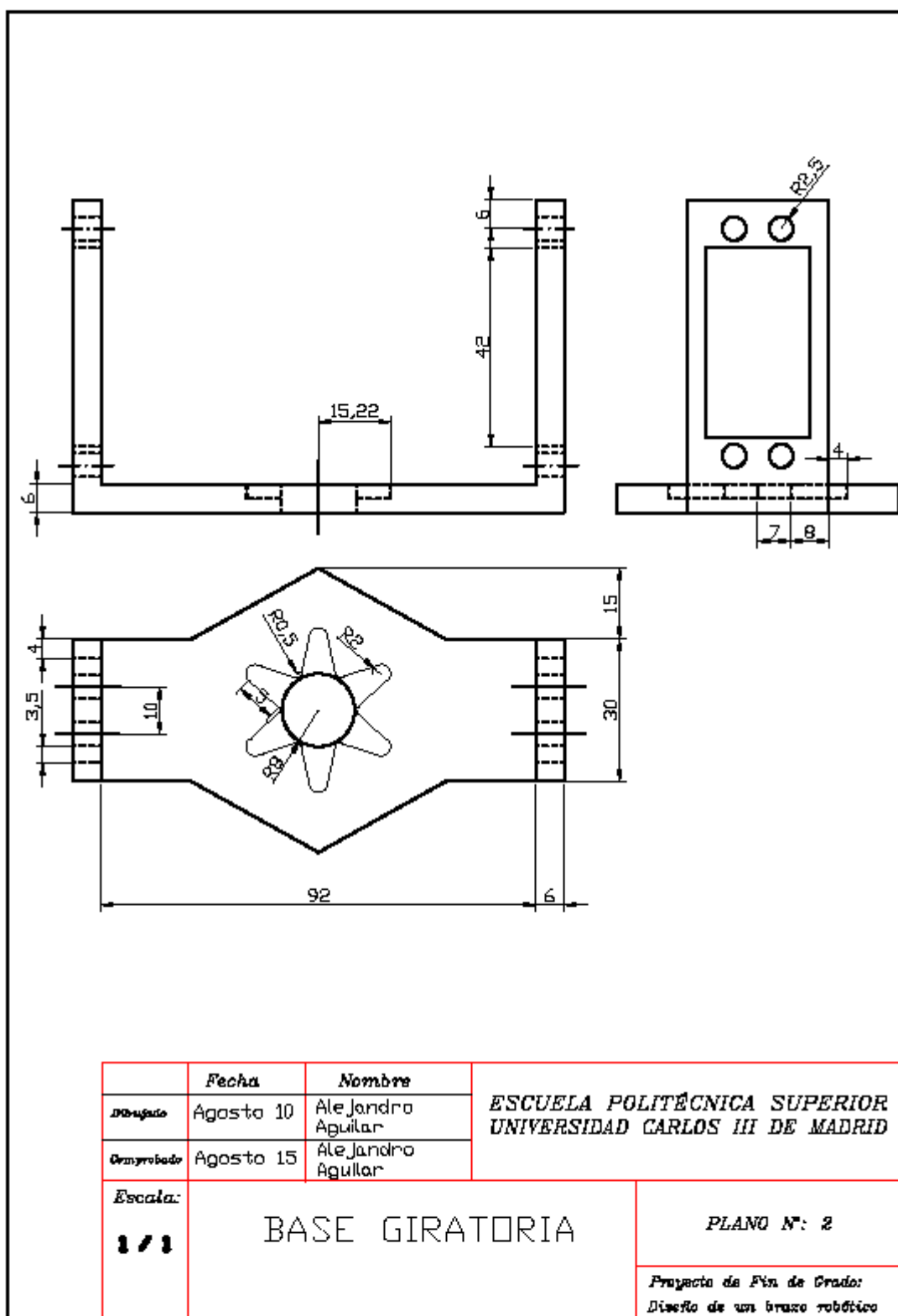
CAPÍTULO 5:

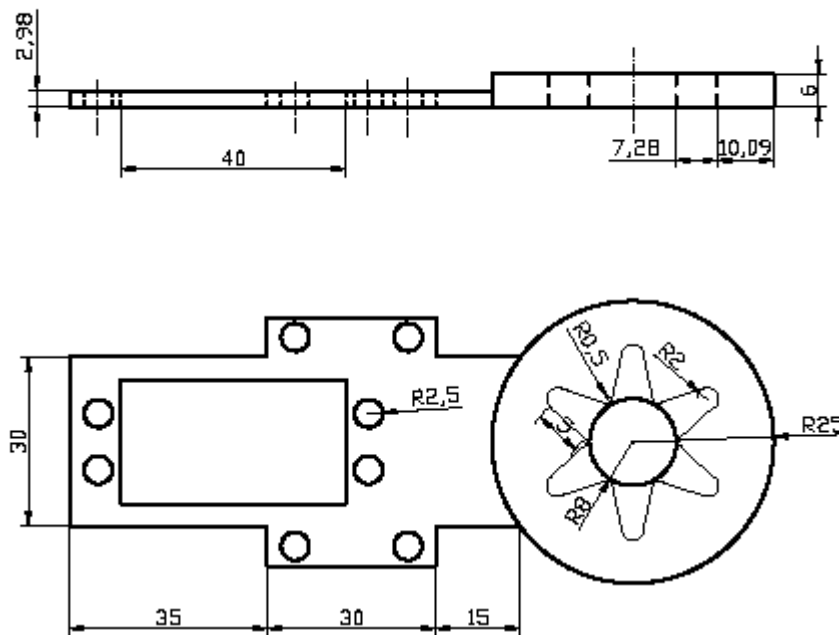
PLANOS Y BIBLIOGRAFÍA

5.1 Planos de las piezas

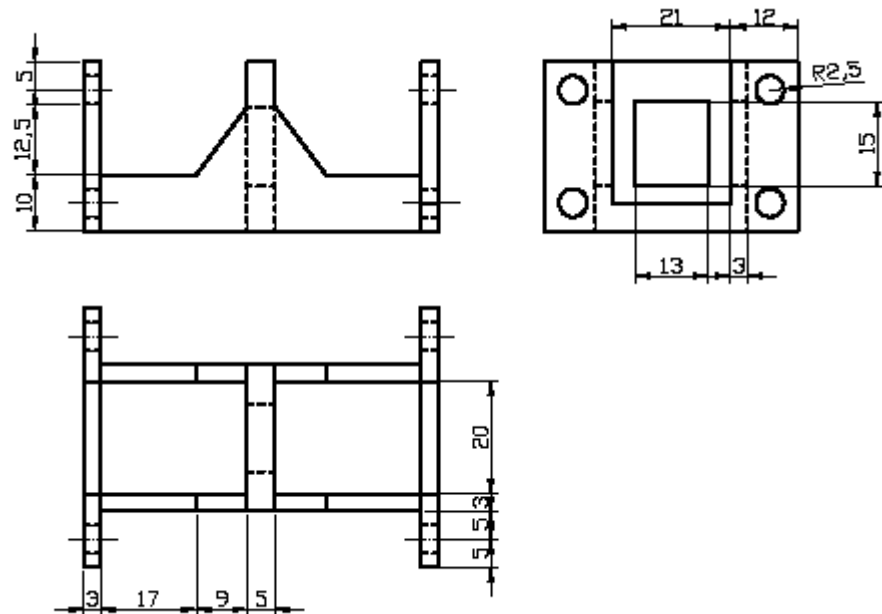


	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Diseñado	Agosto 15	Alejandro Aguilar	
Escala:	BASE PESAS		PLANO N°: 1
1 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico

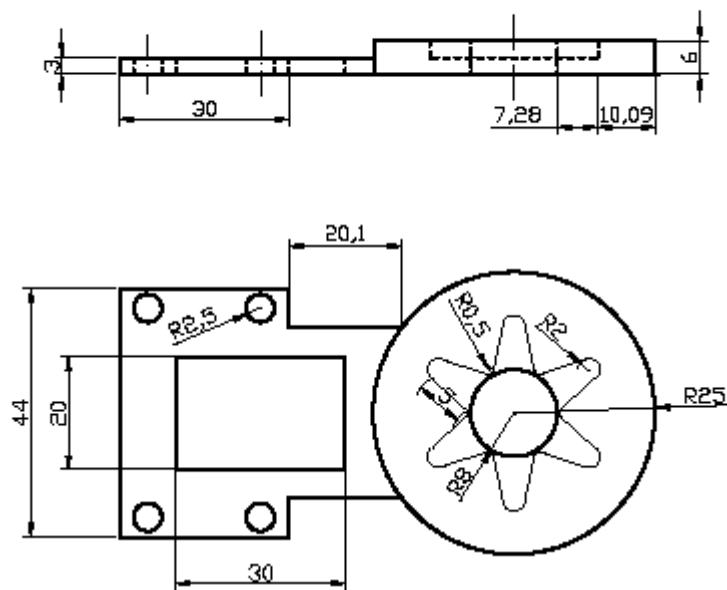




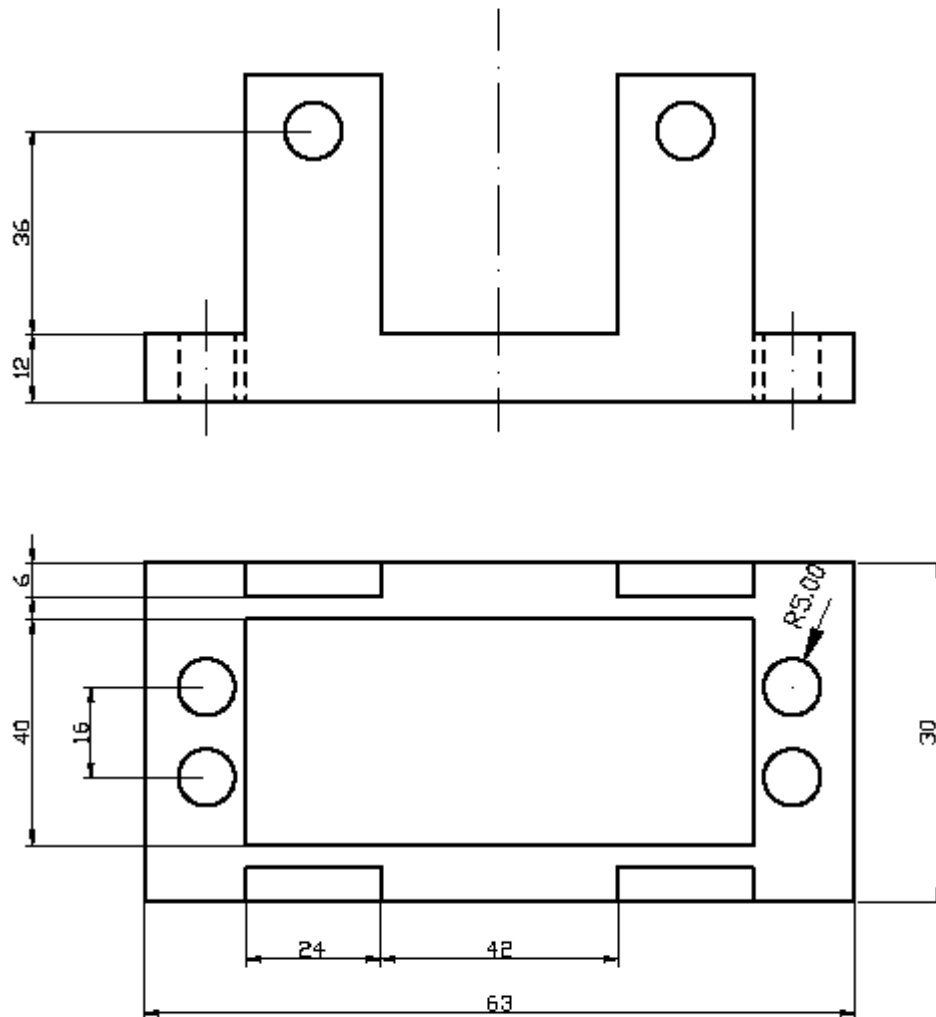
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	ANTEBRAZO		PLANO N°: 3
1/1			Proyecto de Fin de Grado: Diseño de un brazo robótico



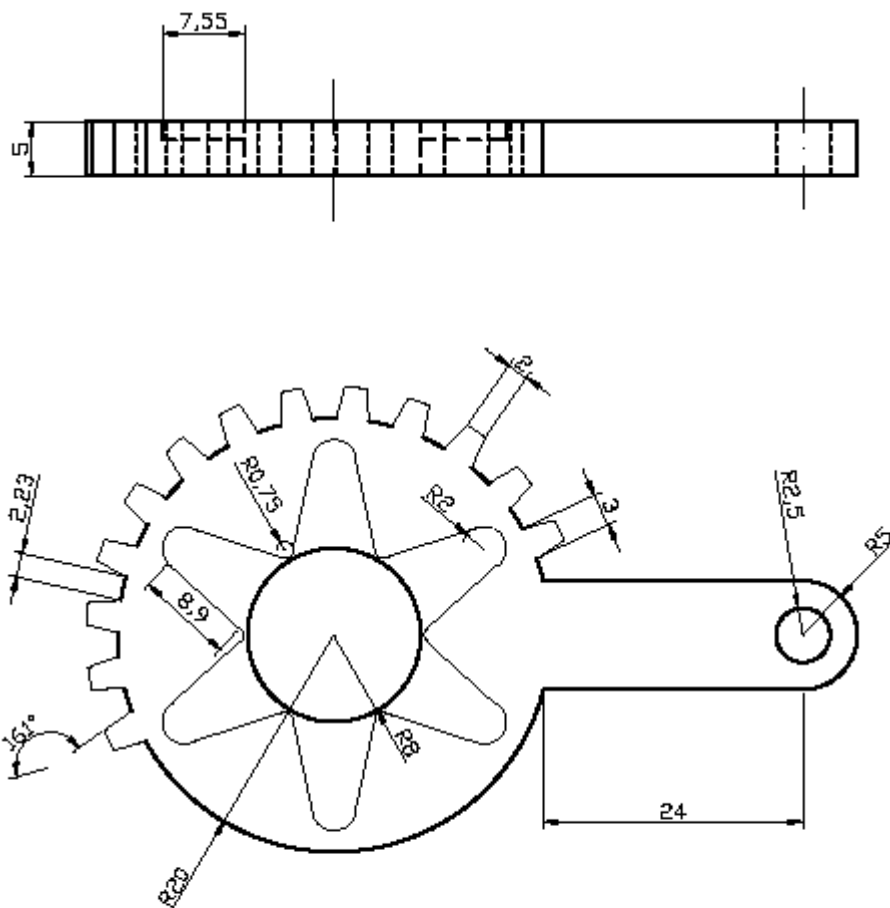
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Diseñado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	PUENTE DEL ANTEBRAZO		PLANO N°: 4
1 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



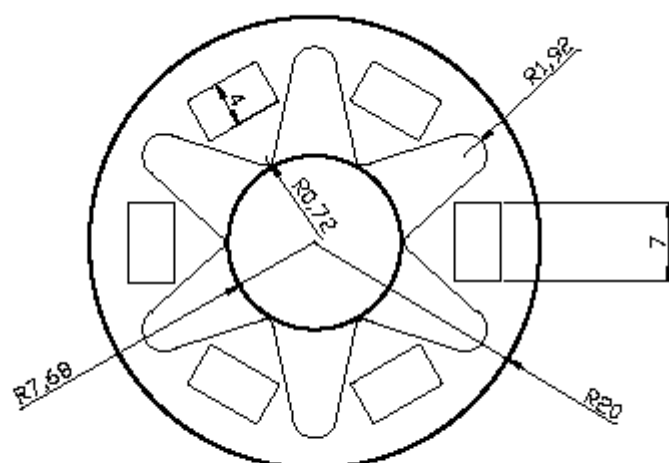
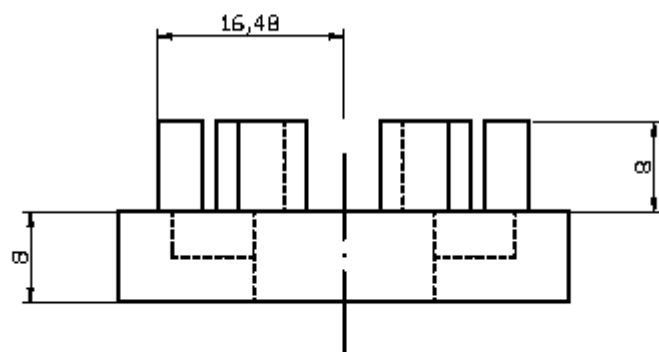
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Diseñado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	BRAZO		PLANO N°: 5
1 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



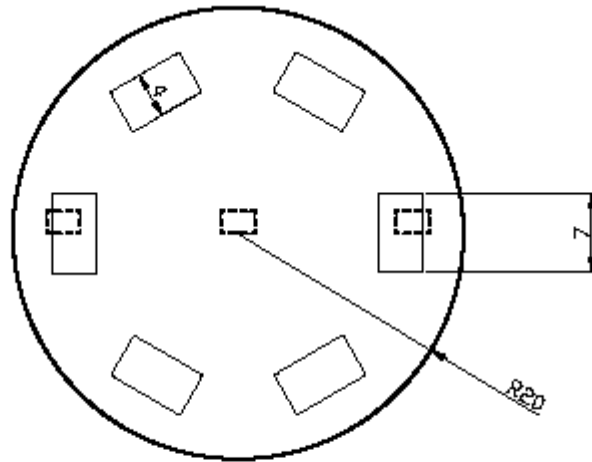
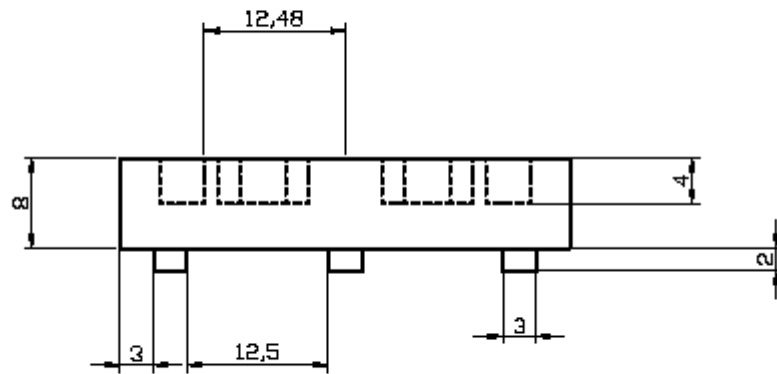
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Diseñado	Agosto 10	Alejandro Aguilar	
Dibujado	Agosto 15	Alejandro Aguilar	
Escala:	PUENTE DEL BRAZO		PLANO N°: 8
8 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



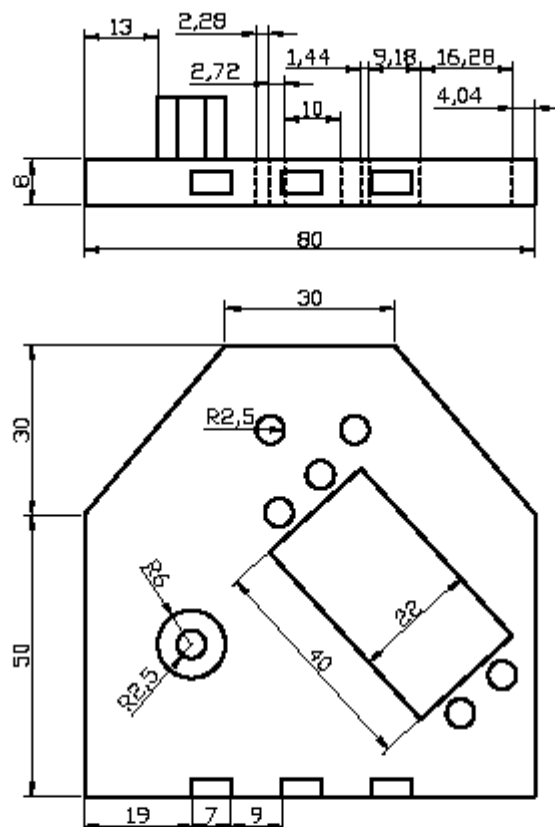
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Corregido	Agosto 15	Alejandro Aguilar	
Escala:	ENGRANAJE DE LA PINZA		PLANO N°: 7
8 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



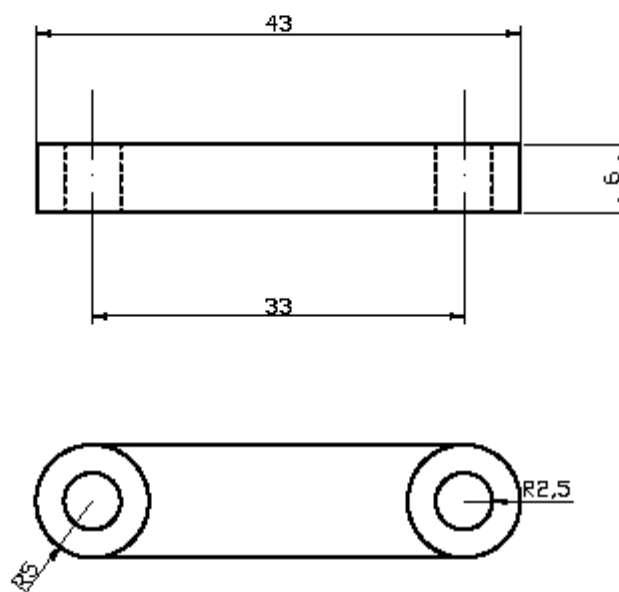
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	MUÑECA CON TAPA		PLANO N°: 8
8 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



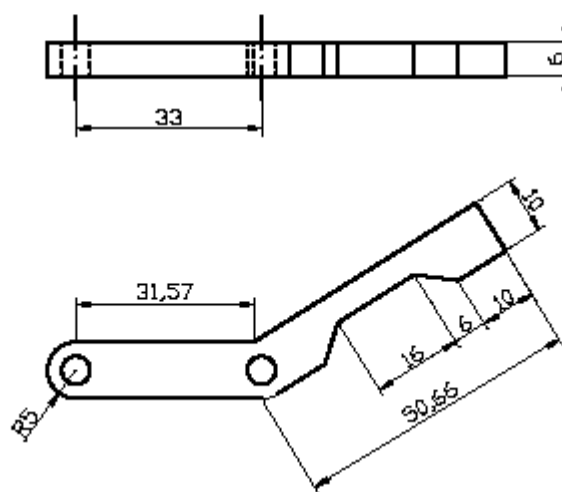
	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Corregido	Agosto 15	Alejandro Aguilar	
Escala:	MUÑECA CON BASE DE PINZA		PLANO N°: 8
8 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Diseñado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	BASE PINZA		PLANO N°: 10
1 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	BARRA DE LA PINZA		PLANO N°: 11
8 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico



	Fecha	Nombre	ESCUELA POLITÉCNICA SUPERIOR UNIVERSIDAD CARLOS III DE MADRID
Dibujado	Agosto 10	Alejandro Aguilar	
Comprobado	Agosto 15	Alejandro Aguilar	
Escala:	AGARRE DE LA PINZA		PLANO N°: 12
1 / 1			Proyecto de Fin de Grado: Diseño de un brazo robótico

5.2 Bibliografía

Creative Commons. "The Open Source Definition", [en línea]. [Agosto 2012]. Disponible en la Web: <http://opensource.org/osd.html/>

Bespoke Prosthetics. "About Bespoke", [en línea]. [Agosto 2012]. Disponible en la Web: <http://www.bespokeinnovations.com/prosthetics/about/about.html>

González, Juan; Prieto-Moreno, Andrés. "Servos Futaba 3003", [en línea]. Septiembre 2009, 13 Agosto 2010, [Agosto 2012]. Disponible en la Web: www.learobotics.com

Estévez Fernández, David. "Hacked Servo", [en línea]. Octubre 2010, Enero 2012, [Agosto 2012]. Disponible en la Web: www.learobotics.com

Economist, The. "3D printing: The printed world." The Economist Newspaper Limited. Feb. 10, 2011. [Agosto 2012]. Disponible en la Web: <http://www.economist.com/node/18114221>

Massachusetts Institute of Technology (MIT). "Licensees of 3D Printing Technology", [en línea]. [Agosto 2012]. Disponible en la Web: <http://web.mit.edu/tdp/www/licensees.html>

"El formato de fichero .stl", [en línea]. [Agosto 2012]. Disponible en la Web: http://www.caddyspain.com/noticias/May06_4.htm

Wolf, Clifford; Kintel, Marius. "OpenSCAD - The Programmers Solid 3D CAD Modeller", [en línea]. [Agosto 2012]. Disponible en la Web: www.openscad.org

New York Times, The. "3-D Printing Spurs a Manufacturing Revolution", The New York Times Newspaper Limited. Septiembre 2010, [Agosto 2012]. Disponible en la Web: <http://www.nytimes.com/2010/09/14/technology/14print.html>

"Arduino Uno", [en línea]. Septiembre 2010, [Agosto 2012]. Disponible en la Web: <http://www.arduino.cc/es>

Berry, David. "Copy, Rip, Burn: The Politics of Copyleft and Open Source". Septiembre 2008. 273 p. ISBN: 0745324142

Juan González. Robots libres e imprimibles. "Convención de Open Source Hardware, Electrónica y Robótica". *Septiembre 2011*.